SEGMENTATION OF CANDIDATE BACILLUS OBJECTS IN IMAGES OF ZIEHL-NEELSEN-STAINED SPUTUM SMEARS USING DEFORMABLE MODELS

PREPARED BY

RONALD DENDERE

This thesis is submitted in partial fulfilment of the academic requirements

for the degree of

Master of Science (Med) in Biomedical Engineering

in the Faculty of Health Sciences

University of Cape Town

February 2009

DECLARATION

I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or materials are indicated in the acknowledgements or are explicitly stated with references as appropriate.

This work is being submitted for the Master of Science (Med) in Biomedical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

_____ _____

Name Date

ACKNOWLEDGEMENTS

The Author takes pleasure in extending his sincere gratitude for the contributions of the following individuals and organisations whose support and input contributed immensely to the progress and outcome of the project.

Firstly, to Associate Professor Tania Douglas, the project supervisor, your contribution is greatly appreciated. Your guidance and advice was an invaluable resource during the course of the project.

The Author also thanks Mr Sriram Krishnan, who provided some useful insights and made several suggestions that contributed to the outcome of the project.

To the members of staff of the National Health Laboratory Services at the Groote Schuur Hospital, in particular, Debbi Viljoen, who prepared the sputum slides and Dr Andrew Whitelaw, who helped with the manual segmentation of the images, thank you for making time in your busy schedules to make a contribution.

The Author thanks Susan Cooper for providing access to the microscopy imaging unit in the Department of Human Biology and for training on image capture; thank you for your assistance.

The Author also thanks the NIH and NRF for funding the project, and the members of staff and the students in the Biomedical Engineering division for their support.

SYNOPSIS

Automated microscopy for the detection of tuberculosis (TB) in sputum smears seeks to address the strain on technicians and to achieve faster diagnosis in order to cope with the rising number of TB cases. Image processing techniques provide a useful alternative to the conventional, manual analysis of sputum smears for diagnosis. In the project described here, the use of parametric and geometric deformable models was explored for segmentation of TB bacilli in images of Ziehl-Neelsen-stained sputum smears for automated TB diagnosis. The goal of segmentation is to produce candidate bacillus objects for input into a classifier.

Watershed segmentation, Canny edge detection and colour thresholding were applied to images to locate objects, for automatic initialisation of the deformable models. The results of the three methods were used in combination to maximise the number of detected objects.

For high-resolution images, objects detected by the Canny edge detector and the watershed algorithms had broken contours, which motivated the use of deformable models as a further segmentation tool. For low-resolution images, however, contours of objects detected by the same algorithms were complete.

To reduce the processing time for deformable model-based segmentation, detected objects were subjected to a discrimination test based on colour and size. An energy minimisation formulation was used to formulate a parametric deformable model. A traditional parametric deformable model with a balloon force was implemented. A variational level set formulation for curve evolution without re-initialisation was used for implementing geometric deformable model-based segmentation.

Validation of deformable model-based segmentation was carried out by manually segmenting bacilli from the images. For each bacillus, the Hausdorff distance between the deformable model-determined border and the manually drawn border was computed and used as a measure of accuracy.

Results indicated that deformable models are able to segment TB bacilli from images of well-prepared sputum smears. However, long processing times could be a limiting factor since the aim of automated TB diagnosis is to achieve faster screening of suspected cases. In addition, under-segmentation of bacilli by deformable models could limit the success of classification algorithms.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The threat posed by tuberculosis (TB) to the world population led the World Health Organisation (WHO) to declare the disease a global emergency in 1993. It is estimated that about a third of the world's population is infected with TB and that about 10% of those infected will suffer from the active disease at some point in their lives (WHO, 2007). TB is the leading cause of death in people living with HIV and AIDS (Harris et al., 2004).

Early detection of active infectious cases is essential to provide patients with adequate treatment to cure the disease and reduce transmission. Sputum smear analysis is the most common diagnostic method for active TB in resource-poor countries (Steingart et al., 2006). Automated microscopy for the detection of TB in sputum smears seeks to address the demand on technicians and to achieve faster diagnosis to cope with the rising number of TB cases. Image processing techniques provide a useful alternative to the conventional manual analysis of sputum smears for diagnosis.

The MRC/UCT Medical Imaging Research Unit is currently running a project to develop an automated microscope for detecting TB bacilli in images of sputum smears stained with a Ziehl-Neelsen (ZN) stain.

This dissertation deals with automatic segmentation of images of sputum smears stained with a ZN stain. Deformable models were applied for segmentation of candidate bacillus objects. Segmented objects may be fed into a classification algorithm that will determine whether or not they are bacilli, based on features such as shape and/or colour.

1.1 Objectives

Published segmentation methods for sputum images have employed traditional edge detection techniques, which produce broken edges in the presence of noise and have used morphological filling to complete broken borders (Veropoulos et al., 1999; Forero et al., 2004).

Deformable models are capable of segmenting objects in noisy images and produce complete contours for objects in such images. Deformable models are a well established segmentation technique in medical image processing and they have been successfully used in a wide variety of applications including segmentation of magnetic resonance and ultrasound images (Xu et al., 2000; Brigger et al., 2000). There is no evidence in the literature to show that

deformable models have been used for segmentation of images of sputum smears. The main objectives of this work can be summarised as follows:

- To examine the use of deformable models as a segmentation method for bacillus-like objects in ZN-stained sputum images.
- To determine the accuracy with which bacilli can be segmented using deformable models.

1.2 Thesis Outline

This report describes the work carried out and the results obtained in an effort to realise the objectives stated above. The second chapter discusses TB, its effects, and the current methods used for diagnosing the disease; this chapter serves to emphasise the need for automated methods for TB diagnosis. Chapter 3 is a review of image processing techniques used by others in the detection of TB and Chapter 4 describes the theory of deformable models.

The description of the methods used for the project has been divided into two chapters. Chapter 5 describes the methods used for detecting objects in images for automatic initialisation of the deformable models. The process of evaluating initially detected objects to determine the eligibility of each object for segmentation by a deformable model is also described in Chapter 5. The implementation and validation of segmentation by deformable models is described in Chapter 6.

Chapter 7 discusses the performance of the algorithms and the results of the validation of the deformable model-based segmentation. Chapter 8 concludes the report and also makes recommendations for improving the methods applied in the project.

1.3 Materials and Tools

Sputum samples used in this project were prepared by the Microbiology section of the National Health Laboratory Services at Groote Schuur Hospital. A total of 40 slides were provided but only 22 slides produced usable images. Images were captured on a Nikon Microphot-FX microscope at the UCT Advanced Light Microscopy Facility in the department of Human Biology. The 40-x and 100-x objective lenses on this microscope are oil immersion lenses and oil was used in capturing all the images.

The images were captured using a Kodak MDS 290 camera that is operated via Adobe Photoshop 7 running on a Pentium 4 desktop computer but focusing is done manually on the

microscope. Images were captured at 40-x and 100-x magnification, and stored with 480 x 720 (low resolution) and 1200 x 1792 (high resolution) pixel resolutions, in 8-bit JPEG format resulting in four sets of images given in Table 1.1.

Table 1.1: Image sets

Image set		Number of images
Magnification	**Resolution (pixel)**	
40-x	480x720	27
	1200x1792	48
100-x	480x720	34
	1200x1792	48

The algorithms were implemented on MATLAB 7.3.0 (2006b) which runs on a Pentium 4 computer with 1GB RAM. The colour thresholding and watershed segmentation algorithms are based on functions in the DIPUM toolbox by Gonzalez et al. (2004). The algorithm for parametric model-based segmentation is based on functions from the GVF toolbox by Xu et al. (1999). The algorithm for geometric model-based segmentation is based on the demonstration code for level sets without re-initialisation by Li et al. (2005).

2. TUBERCULOSIS

This chapter gives a brief description of tuberculosis (TB), the threat it poses and the limitations of the current methods used in its diagnosis.

2.1 Tuberculosis

Tuberculosis is an infectious, air-borne, respiratory disease caused by a micro-organism called *Mycobacterium tuberculosis*. TB has two stages, namely infection and active disease. A person who has a TB infection carries the TB bacteria, but the immune system has the bacteria under control. A person with TB infection will not have the symptoms of the disease and cannot spread the disease.

The second stage is active TB. TB infection progresses into active TB when the TB bacilli overcome the immune system and begin to reproduce. A person with active TB can spread the disease and will exhibit the symptoms of the disease that include chest pains, a persistent cough lasting for at least three weeks, weight loss, fever, coughing up blood, breathlessness, loss of appetite and night sweats (SANTA, 2004). Active TB is divided into pulmonary and extra-pulmonary TB. Pulmonary TB attacks the lungs and is the more common of the two. Extra-pulmonary TB results when the bacteria has spread to other parts of the body, affecting, among others, the nervous system, lymph nodes, genitals, kidneys and even the skin (Todar, 2008).

According to the World Health Organisation (WHO) about a third (1.7 billion) of the people in the world are infected with TB and about 5-10 % of those infected will suffer from active TB at some point in their lives. The WHO estimates that 1.6 million people died of TB in 2005 and that the number of new cases is increasing each year. In April 1993 the WHO declared TB a global emergency and in 2006 the organisation launched the new Stop TB Strategy (WHO, 2007).

The South African National Tuberculosis Association (SANTA) estimates that the TB incidence in the country is 1084 cases per 100 000 population. About 66.4% of these cases are patients who are HIV positive. It is also estimated that at least 66% of the South African population is infected with TB (SANTA, 2004).

TB is the main cause of death for people living with HIV and AIDS. The WHO estimates that in countries with high rates of HIV, up to 80 % of people infected with TB are HIV positive

and that TB prevalence and mortality is increasing in sub-Saharan Africa because of the persistence of HIV in the region. The steady increase in TB incidence is straining diagnostic services (Harris et al., 2004) and therefore there is a great need for faster and more efficient diagnosis methods for TB.

2.2 Sputum Smear Test for Diagnosis of TB

While active TB can also be diagnosed by a chest X-ray, the treatment strategy recommended by the World Health Organisation relies heavily on the results of sputum smear microscopy (Nguyen et al., 1999). In this method, a sputum sample is taken from a patient and it is spread on a microscope slide. The sample is stained with either a Ziehl-Neelsen (ZN) or an auramine stain. In the ZN staining method, a red dye is applied to a sputum smear on a test slide under heat, leaving all present bacilli stained reddish-pink. An acid-alcohol is used to wash the test slide; the stained bacteria retain the dye and are referred to as acid-fast bacilli. A blue dye is applied to the test slide; only acid-fast bacilli retain the reddish-pink stain while the rest of the slide contents turn blue. With the auramine staining method, the bacilli become fluorescent in a black background. Figure 2.1 illustrates the difference in images of sputum samples stained by the two methods.

a) Image of an auramine-stained sample (Source: www.lung.ca/tb/images/)

b) Image of a ZN-stained sample

Figure 2.1: Staining methods for sputum smears.

A bright-field microscope is used for viewing ZN-stained sputum samples whereas a fluorescence microscope is used for auramine-stained samples. Fluorescence microscopes are complex and quite expensive, to purchase as well as to maintain; for this reason, the ZN staining method is widely used in developing countries.

In addition to TB bacilli, a sputum sample may also contain other foreign bodies and other bacteria. The bacilli measure approximately 1 to 10 μm in length and 0.2 to 0.6 μm in width and they can present a straight, curved or bent shape. Individual bacilli may display heavily stained areas and zones of alternating stain producing a beaded appearance (Collins et al., 1985).

2.3 Automated TB Diagnosis

Developing countries are faced with shortage of skilled technicians, inadequate equipment and rising TB cases. Recommendations for smear analysis require that 100 high-power fields be viewed per slide (Sadaphal et al., 2008). Manual screening for TB bacilli in sputum smear analysis is labour intensive and the false negative rate is high (Forero et al., 2004). For these reasons a small number of researchers have channelled their efforts into developing automated detection techniques to reduce the strain on technicians and reduce the false negative rate.

Forero et al. (2004) and Veropoulos et al. (1999) developed automatic detection methods that detect TB bacilli in images of auramine-stained sputum smears. The techniques developed have many advantages as discussed later but the fluorescence microscope for which their methods are designed is currently not in widespread use in developing countries.

Veropoulos (2001), Russell (2006) and Sadaphal et al. (2008) developed image processing algorithms for automated TB detection in ZN-stained sputum samples. These methods are discussed in Section 3.3.

2.3.1 Advantages of Automated Bacillus Detection

Automated TB diagnostic methods bring about several benefits in the fight against the disease. Some of the advantages of computer-based diagnosis are:

- Computer-assisted diagnosis can be faster than manual screening.
- A large number of slides can be analysed and hence the number of patients tested in any given amount of time is increased.
- Human involvement in the screening process is reduced. This will result in three desirable effects; less experienced staff can conduct the screening, labour costs are reduced and technicians suffer less fatigue, since the need to view the slides on the microscope is eliminated or reduced.

- Accuracy may be improved; in areas of high TB incidence like sub-Saharan Africa where a large number of patients have to be tested, technicians may test slides less thoroughly than required and smear positive cases may go unnoticed. Automated detection would ensure that the entire slide is examined in a short space of time and would thus reduce cases of misdiagnosis.

In essence, an automated TB detection process may allow a greater number of patients to be tested at a faster rate by fewer technicians and with higher accuracy.

3. LITERATURE REVIEW

This chapter summarises the material that forms the background to the project, providing an introduction to image segmentation and an overview of the segmentation of TB bacilli by other researchers.

3.1 Image Representation

A digital image is composed of a rectangular array of pixels. Digital images can be represented as gray-level or colour images. In a gray-level image, also called an intensity image, the brightness of a pixel is represented by the intensity value of the image at the pixel location. The intensity values range from 0, which represents black, to a maximum value, which represents white. The maximum gray-level value for an *n*-bit system is $2^n - 1$. Intensity values between 0 and a maximum value represent shades of gray that increment towards white.

Colour images are a combination of intensity images, which are interpreted differently by different colour models. A colour model is a specification of a coordinate system and a subspace within that system where each colour is represented by a point; the RGB (red, green, blue) and the HSI (hue, saturation, intensity) are the most common colour models (Gonzalez and Woods, 2002).

3.1.1 The RGB Colour Space

The RGB colour space is based on the principle that any colour is a combination of the three primary colours; red, green and blue. A colour image is composed of three constituent images, each containing the intensity values of each primary colour at each pixel location. The colour of each pixel can be represented by a point on or in the cube shown in Figure 3.1; the point is defined by a vector extending from the origin (Gonzalez and Woods, 2002). Pixels that lie on the indicated diagonal vary from black to white.

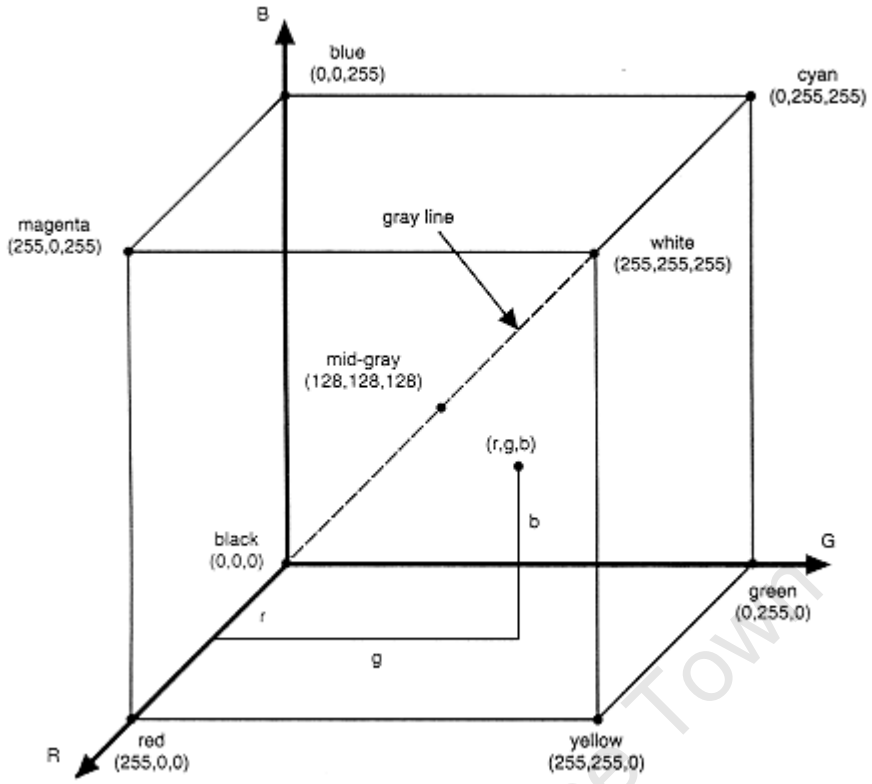


Figure 3.1: The RGB colour space.

3.1.2 The HSI Colour Space

In the HSI colour space, a colour image has three component images. The colour information is contained in the hue and saturation component images. The intensity component image specifies the overall brightness of pixel locations, regardless of colour (Castleman, 1996). The hue, the basic colour, and saturation, purity of the colour, parameters are defined on a circle; the hue is expressed as an angle, with red as the reference angle and the saturation is expressed as the radius from the origin of the circle. The intensity is specified on a vertical axis so that the colour space takes a cylindrical form (Gonzalez and Woods, 2002).

3.2 Image Segmentation

The major steps in identifying objects in images, which is the problem dealt with in automated TB detection, can be summarised as follows:

1. Image input
2. Pre-processing

3. Segmentation
4. Feature selection or extraction
5. Object classification
6. Output

Image segmentation can be defined as the process of separating objects in an image into distinct, homogenous regions to identify objects and borders for object description and recognition (Gonzalez and Woods, 2002).

Segmentation is an important step in image processing which aids extraction of information and attributes from images for image understanding and interpretation. Its fundamental roles are to allow quantification of regions of interest and to reduce the dataset by focusing the quantitative analysis on the extracted regions of interest. It is often a difficult process due to factors such as poor spatial resolution and noise (Gonzalez and Woods, 2002). Segmentation techniques vary according to type of image. While manual segmentation is the simplest, it can be time-consuming and tedious. The simplest segmentation techniques can be divided into the following classes:

- ❖ Thresholding techniques
- ❖ Edge-based segmentation
- ❖ Region-based segmentation

These methods, unlike deformable models, do not require a model to guide the segmentation process.

3.2.1 Thresholding Methods

In a thresholding technique, a predefined value (threshold) is chosen and the image under analysis is divided into two pixel groups; a group of pixels that have intensity values that are greater than or equal to the threshold and a group of pixels that have intensity values that are below the threshold. For an input image, *f*, the thresholding operation produces a binary image, *I*, that can be defined as:

$$I(i,j) = \begin{cases} 1 & for\ f(i,j) \geq T \\ 0 & for\ f(i,j) < T \end{cases} \qquad (\mathbf{3.1})$$

where T is the threshold and i and j are integers that describe pixel locations on a Cartesian plane.

Global thresholding is a thresholding technique in which a single threshold value is applied to an entire image. It is simple and computationally fast. It performs well if an image contains objects with homogeneous intensity and/or the contrast between the objects and the background is high (Wong, 2005). The method may fail to produce usable results when two or more structures have overlapping intensities.

Other thresholding techniques exist which may be classified as local or dynamic thresholding. These methods can be useful in cases where a single threshold value cannot be determined for an entire image or where a single threshold does not give good results.

3.2.2 Edge-based Segmentation Methods

An edge may be defined as a collection of connected pixels that lie on the boundary between two homogeneous regions of different intensities (Wong, 2005). Segmentation techniques based on edges have two stages of operation; edge-detection and edge linking. Edge detectors convolve an image with a mask that runs through the entire image to yield an edge magnitude image. A thresholding operation is then carried out on the edge magnitude image to determine the location of edges. Edge detectors use either first-order or second-order gradient information of an image. First-order derivatives have local minima and maxima at edges while second-order derivatives have zero crossings at the edges.

After edges have been detected, edge-linking is performed to link edge pixels in order to produce closed regions. False edge pixels may result in noisy images. Examples of edge detectors include the Sobel, Prewitt, Kirsch and Canny edge detectors, each of which uses a unique mask.

3.2.3 Region-based Segmentation Methods

Region-based segmentation methods examine all pixels in an image and form separate regions by joining neighbouring pixels that have similar properties based on a predefined similarity criterion (Gonzalez and Woods, 2002). Region growing is an example of a region-based segmentation method in which a user selects a starting pixel, a seed, that belongs to a target region of interest. The algorithm then inspects all neighbouring pixels for similarity according to a predefined similarity measure. Pixels that satisfy the similarity criterion are

added to the region where the seed belongs. The process is repeated until no pixel can be assigned to any region.

Region splitting is another example of a region-based segmentation method that works in an opposite fashion to region growing. The algorithm starts with the entire image as a region and splits the image in accordance with a similarity criterion. A disadvantage associated with region-based methods is that regions are not necessarily split on natural borders of objects and it is therefore difficult to accurately determine object borders (Kovacevic et al., 1999).

3.3 Segmentation of TB Bacilli in Images of Sputum Smears

Image processing techniques provide a useful alternative approach for TB diagnosis to the conventional manual analysis of sputum samples. The image processing techniques, particularly for segmentation, that have been explored for TB bacillus detection in sputum images are discussed in this section.

The algorithms applied by Veropoulos (2001) and Sadaphal et al. (2008) for recognising acid-fast bacilli in images captured from ZN-stained smears have a similar structure and can be divided into three stages; image segmentation, object measurement, and classification. Russell (2006) only focused on segmentation of bacilli from ZN-stained sputum smear images.

In the algorithm developed by Veropoulos (2001), an input image was converted from the RGB colour space to the HSI colour space. Segmentation was achieved by thresholding the saturation component image using its average intensity value. The Canny edge detector was used to detect edges and this was followed by a region labelling operation. In the region labelling step, the regions identified by the Canny edge detector were assigned a unique intensity value and regions with intensities that were out of a predefined range were discarded. For reliable object recognition, gaps in the detected contours were filled by edge pixel linking. The inner boundary of each region was traced and information about the position of boundary pixels was gathered, and shape descriptors were computed from boundary curves of each detected region. The shapes of the detected objects were described by Fourier descriptors and compactness, while their colours were described by the average, standard deviation and the centroid of the RGB values of the region covered by each object; a kNN classifier was used to classify the detected objects using these features.

In the segmentation method of Sadaphal et al. (2008), pixel probabilities were derived from a 3-D probability density function histogram that was constructed by manually segmenting bacilli from the images. A binary mask with 1s in the pixel positions that exhibit the required bacillus colour was created by thresholding the pixel probabilities. The binary mask was morphologically dilated using a circular structuring element.

By treating bacilli as generally having an elliptical shape, the shape of detected objects was described using the eccentricity and the axis ratio. Objects whose eccentricity and axis ratio values were below threshold values were labelled as non-bacillus objects. For objects that met the eccentricity and axis ratio criteria, the object size was computed. Objects whose size was outside the range $mean\ size \pm 1.5\sigma$, where σ is the standard deviation, were labelled as possible bacilli while those whose size was within the required were range labelled as definite bacilli.

Russell (2006) examined the use of gray-level (crisp and fuzzy thresholding) and colour (Canny edge, colour edge extraction and colour compensation thresholding) segmentation techniques for segmenting bacilli from sputum images. Colour compensation thresholding yielded the best results, segmenting only TB bacilli in the images.

Colour compensation thresholding involves the use of colour compensation techniques followed by crisp thresholding. The concept behind colour compensation is that, for the RGB colour system, the brightness of a colour component is spread among all three channels (Castleman, 1998). A matrix that specifies the spread of each colour component in each channel is experimentally determined and each element of the matrix represents the proportion of the brightness of each component in each channel. Russell (2006) came up with an average matrix that took into account the colour of the bacilli to specifically segment TB bacilli from the images. The segmented objects had broken borders but no additional processing was carried out to fill the breaks.

Fluorescence microscopy for TB detection in auramine-stained sputum smears provides better performance than light microscopy for detection in ZN-stained smears, and allows the detection of more positive smears (Hanscheid, 2008). Manufacturers of microscopes are making efforts to find low-cost alternative methods for manufacturing fluorescence microscopes (Hanscheid, 2008); this may allow widespread application of fluorescence microscopy for TB detection in resource-poor countries.

Image processing techniques have been applied to identify TB bacilli in images of sputum smears stained with an auramine stain (Veropoulos, 2001; Forero et al., 2004). Veropoulos (2001) applied the same sequence of steps described above for ZN-stained smears for the detection of bacilli in images of auramine-stained smears.

Forero et al. (2004) used the Canny edge detector to detect edges in images. A morphological closing operation was applied to fill the gaps in the detected borders. To eliminate edges with unconnected ends, all closed contours were filled out and then a morphological operation was applied. Objects were rejected based on their intensity values in the red and green channels of the RGB colour space. Objects were further eliminated based on the perimeter of their borders. The remaining objects were classified as either bacillus or non-bacillus based on compactness, eccentricity and moments.

4. DEFORMABLE MODELS: THEORY

4.1 Introduction

Originally suggested by Kass et al. (1987), the use of deformable models for image segmentation has been an area of extensive research. Deformable models are a useful technique in medical image processing (Brigger et al., 2000) because of superior performance compared to traditional, model-free methods in the presence of noise. Deformable models have been used for a wide range of applications including medical image processing, video tracking and stereo matching.

A deformable model is regarded as a model-based segmentation technique that employs an analytical model to describe the shape of an underlying region of interest (Wong, 2005). Researchers use different terms to refer to this technique with names such as active contours, deformable models, deformable templates, and snakes being common in the literature. The terms, deformable models and snakes are used interchangeably in this report.

Xu and Prince (1998) have defined deformable models as curves or surfaces defined within an image domain that can move under the influence of internal forces generated from within the model and external forces computed from the image data so that the model will conform to object boundaries.

Deformable models are broadly classified into parametric or geometric deformable models based on representation and implementation (Han et al., 2003). Parametric deformable models are represented explicitly as parameterised curves; this is referred to as the Lagrangian formulation. Geometric deformable models are represented implicitly as level sets of 2-D distance functions; this is referred to as the Eulerian formulation. The flow chart in Figure 4.1 illustrates the general working principle for segmentation using a deformable model.

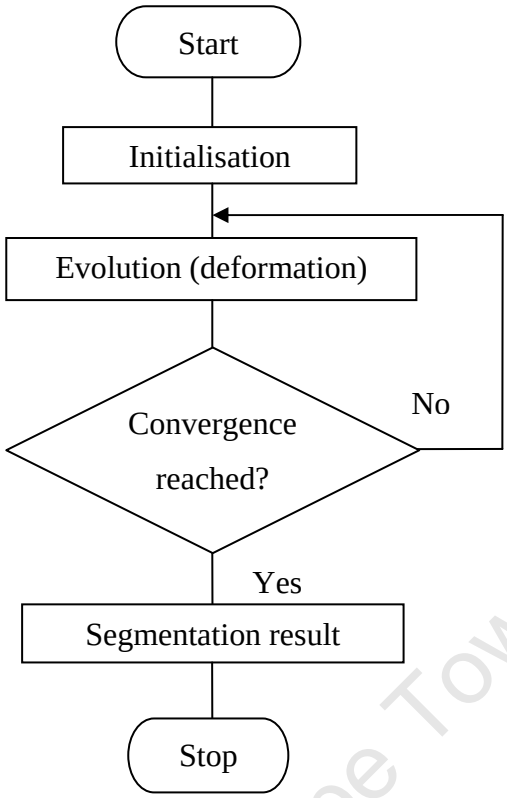


Figure 4.1: Flow chart for deformable model-based segmentation.

Initialisation of the deformable model involves specifying a starting state in terms of position and shape of the deformable model on the image being analysed. All the model parameters are specified during initialisation. After initialisation, the model then deforms towards an object border by solving an evolution partial differential equation (PDE). Evolution of the deformable model converges when the model reaches a border and the model cannot deform any further; at this point the position of the snake represents the detected border.

4.2 2-D Parametric Deformable Models

A parametric deformable model can be represented mathematically as:

$$\boldsymbol{V}(s) = \big(x(s), y(s)\big)\ s \in [0,1] \qquad (\mathbf{4.1})$$

where *s* can be any parameter.

The model is represented by two vectors that contain the coordinates of points on the snake. The formulations for parametric deformable models given below are derived from Xu et al. (2000).

4.2.1 Energy Minimisation Formulation of Parametric Deformable Models

The focus of this method is to find a parametric curve that minimises the weighted sum of the internal energy of the deformable model and the external energy derived from image data. Minimising the total energy yields internal forces and external forces. After initialisation the curve is then forced to move under the influence of both forces to find object boundaries. The energy function to be minimised is of the form:

$$\varepsilon(\boldsymbol{V}) = S(\boldsymbol{V}) + \mathcal{P}(\boldsymbol{V}) \qquad (\mathbf{4.2})$$

where $S(\boldsymbol{V})$ is the internal energy and is defined as:

$$S(\boldsymbol{V}) = \frac{1}{2}\int_0^1 \alpha \left|\frac{\partial \boldsymbol{V}}{\partial s}\right|^2 + \beta \left|\frac{\partial^2 \boldsymbol{V}}{\partial s^2}\right|^2 ds \qquad (\mathbf{4.3})$$

The first term of the integral discourages stretching and the second term discourages bending. The combination of the terms makes the curve behave like an elastic, rigid rod. α and β control the curve's elasticity and rigidity respectively and are referred to as regulating parameters.

The second term of Equation 4.2 is the potential (external) energy functional that is obtained by integrating a potential energy function $P(x, y)$ along the contour $\boldsymbol{V}(s)$:

$$\mathcal{P}(\boldsymbol{V}) = \int_0^1 P(\boldsymbol{V}(s)) \qquad (\mathbf{4.4})$$

$P(x, y)$ is derived from the image data and has its minimum values at object boundaries. A Gaussian potential force is then derived from the potential energy function and a snake is designed to deform to minimise the total energy.

Potential forces used by traditional snakes can be defined as the negative gradient of a potential energy function while non-potential external forces cannot be expressed as a negative gradient of a potential energy function. In grey-level images, a typical potential energy function designed to lead a deformable model to an edge is:

$$P(x, y) = -\omega_e |\nabla [G_\sigma(x, y) * I(x, y)]|^2 \qquad (\mathbf{4.5})$$

where w_e is a positive weighting parameter, $G_\sigma(x, y)$ is a 2-D Gaussian function with standard deviation σ, * is the 2-D convolution operator, ∇ is the gradient operator and $I(x, y)$ is the image function.

Increasing the standard deviation increases the capture range for the model, however if σ becomes too large, significant blurring of the edges may occur, causing a shift in the boundary location, resulting in less accurate results. The potential energy can therefore be calculated using different values for σ to find the most suitable value; the procedure for minimising the energy functional is the same.

A deformable model that minimises $\varepsilon(\boldsymbol{V})$ must satisfy the Euler equation given by Equation 4.6.

$$\alpha \left|\frac{\partial \boldsymbol{V}}{\partial s}\right|^2 - \beta \left|\frac{\partial^2 \boldsymbol{V}}{\partial s^2}\right|^2 - \nabla P(\boldsymbol{V}) = 0 \qquad (\mathbf{4.6})$$

Equation 4.6 can be considered as a force balance equation between the internal and external forces where the internal force is given by Equation 4.7 and the external potential force is given by Equation 4.8.

$$F_{int}(\boldsymbol{V}) = \alpha \left|\frac{\partial \boldsymbol{V}}{\partial s}\right|^2 - \beta \left|\frac{\partial^2 \boldsymbol{V}}{\partial s^2}\right|^2 \qquad (\mathbf{4.7})$$

$$F_{ext}(\boldsymbol{V}) = -\nabla P(\boldsymbol{V}) \qquad (\mathbf{4.8})$$

To solve Equation 4.6, the deformable model is made dynamic by treating $\boldsymbol{V}$ as a function of time, *t*, and s. The partial derivative of $\boldsymbol{V}$ with respect to *t* is then equated to the left hand side of Equation 4.6:

$$\frac{\partial \boldsymbol{V}}{\partial t} = \alpha \left|\frac{\partial \boldsymbol{V}}{\partial s}\right|^2 - \beta \left|\frac{\partial^2 \boldsymbol{V}}{\partial s^2}\right|^2 - \nabla P(\boldsymbol{V}) \qquad (\mathbf{4.9})$$

When the solution $\boldsymbol{V}(s, t)$ stabilises, the right hand side of Equation 4.9 disappears and a solution of Equation 4.6 is achieved. A numerical solution of Equation 4.9 is found by making it discrete and iteratively solving the discrete system.

4.2.2 Dynamic Force Formulation of Parametric Deformable Models

In this type of formulation, the deformable model is formulated directly from a dynamic force formulation:

The dynamics of an evolving model, $\boldsymbol{V}(s,t)$, must satisfy:

$$\mu\frac{\partial^2 \boldsymbol{V}}{\partial t^2} = F_{damp}(\boldsymbol{V}) + F_{int}(\boldsymbol{V}) + F_{ext}(\boldsymbol{V}) \qquad (\mathbf{4.10})$$

where μ is a coefficient that has a mass unit, $F_{int}(\boldsymbol{V})$ is the internal force, $F_{ext}(\boldsymbol{V})$ is the external force and $F_{damp}(\boldsymbol{V})$ is the damping force defined by Equation 4.11, in which γ is the damping coefficient.

$$F_{damp}(\boldsymbol{V}) = -\gamma\frac{\partial \boldsymbol{V}}{\partial t} \qquad (\mathbf{4.11})$$

For image segmentation, μ is set to zero and the dynamics of the contour become

$$\gamma\frac{\partial \boldsymbol{V}}{\partial t} = F_{int}(\boldsymbol{V}) + F_{ext}(\boldsymbol{V}) \qquad (\mathbf{4.12})$$

The internal forces are the same as given in Equation 4.7 while the external forces do not strictly have to be potential forces. The external force can be expressed as a sum of various forces which can be implemented simultaneously to increase the capture range for the snake. The capture range of an external force is the maximum distance from an object border at which the force is strong enough to attract an initial model and pull it towards the border. The various types of forces are discussed below.

i) Multi-scale Gaussian Potential Force

This type of force was introduced as the external force for the energy minimisation formulation. When using the Gaussian potential force, the width (σ) of the Gaussian must be small enough that the model can follow the desired boundary accurately. This, however, comes at the cost of reducing the capture range. To solve this problem, Kass et al. (1987) proposed using different Gaussian potential forces to increase the capture range and maintain the localisation accuracy. Their method was to initially use a large value for σ and therefore increase the capture range. As the model approaches equilibrium, the value of σ is reduced so that the deforming model can accurately conform to the desired boundary. Even though this

method increases the capture range for the external force, there is no proven criterion when σ should be changed (Xu et al., 2000).

ii) Pressure Force

This type of force was introduced by Cohen (1991). He used a pressure force in addition to the Gaussian force to increase the capture range for deformable models. He referred to deformable models that use pressure forces as balloons since the pressure force can either inflate or deflate the model.

iii) Distance Potential Force

Cohen and Cohen (1993) proposed another technique to increase the capture range by defining a distance map. The value of the distance map at each pixel is obtained by calculating the distance between the pixel and the closest boundary point based on the Euclidean distance. Using this technique, a potential force field with a large capture range can be obtained.

iv) Dynamic Distance Force

This external force is similar to the distance potential force but does not have difficulties progressing into boundary concavities. The external force is derived by calculating a signed distance at each point on the model by determining the closest boundary point. The force, referred to as the dynamic distance force, has a fairly long capture range.

v) Interactive Force

Kass et al. (1987) proposed two kinds of interactive forces namely spring and volcano forces. These forces are used in situations where the user needs to control the model as it deforms. Spring forces are defined to be proportional to the distance between a fixed point, *V*, on the model and a user specified point, *p*, as given by Equation 4.13. The forces will therefore act to pull the snake toward the point.

$$F_s = w_s(p - V) \qquad (\mathbf{4.13})$$

where w_s is a weighting parameter for the spring force.

Volcano forces on the other hand are designed to push away the snake from a local region around a volcano point, *p*. The force is computed in a neighbourhood $\mathcal{N}(p)$ as follows:

$$F_v = \begin{cases} w_v \dfrac{r}{|r|^3} & V \in \mathcal{N}(p) \\ 0 & V \notin \mathcal{N}(p) \end{cases} \qquad (\mathbf{4.14})$$

where $r = V - p$ and w_v is a weighting parameter.

4.2.3 B-splines and B-snakes

Generally, a spline is considered to be a piecewise parametric curve. Splines are popular in computer graphics because of the simple methods used for constructing them and their capability to approximate complex shapes through curve-fitting (Rogers and Adams, 1990). A spline consists of spline points, also called knots, and a curve that interpolates the spline points. The spline shape is controlled by the spline coefficients, *c(k)*, also called control points. For B-splines, the knots are joined by short, straight lines called basis functions and the interconnected basis functions form the entire curve. The spline points are directly related to the control points through a system of linear equations (de Boor, 1978). B-splines are continuous functions and they are often used to build parametric curves. Cubic B-splines, splines of degree 3, are often used because the continuity at this degree provides implicit smoothness to adjacent points of the corresponding parametric curve.

A uniform cubic B-spline curve is parametrically built on a set of coefficients, *c(k)*:

$$g(s) = \big(x(s), y(s)\big) \qquad (\mathbf{4.15})$$

where

$$x(s) = \sum_{k=0}^{n-1} c_x(k) B_k(s) \qquad (\mathbf{4.16})$$

and

$$y(s) = \sum_{k=0}^{n-1} c_y(k) B_k(s) \qquad (\mathbf{4.17})$$

c_x and c_y are the Cartesian plane coordinates of the coefficients, $B_k(s)$ is the basis function centred on the k^{th} coefficient and *n* is the number of coefficients.

The B-spline-based snake, referred to as a B-snake, was first introduced by Menet et al. (1990) as an alternative to traditional snakes. B-snakes combine snake evolution with B-spline representation for curves.

Parametric B-spline representation for parametric deformable models was introduced to solve some of the problems associated with traditional parametric models. Menet et al. (1990) linked the energy minimisation formulation proposed by Kass et al. (1987) with B-spline representation of snake curves by integrating the α and β coefficients in Equation 4.3 and the first and second order derivatives of the B-spline function into a single matrix during numerical implementation. The matrix is used in a pair of equations that are solved iteratively for the *x* and *y* coordinates of points on the curve.

This technique allows compact representation of the curve that is characterised by fewer parameters and smoothness is implicitly built into the model (Brigger et al., 2000). While the curve for a traditional parametric deformable model is represented as an ordered collection of discrete points, the curve for a B-snake is described continuously and parametrically using B-splines. Initialisation and evolution can be either one of the methods described in Sections 4.2.1 or 4.2.2, the only difference is the representation of the curve.

4.2.4 Numerical Implementation of Parametric Deformable Models

Various numerical implementations of parametric deformable models have been reported in the literature. Finite difference methods, dynamic programming and greedy algorithms have been used for this purpose. The finite difference method is preferred because it is simple and easy to compute (Xu et al., 2000).

4.2.5 Advantages of Parametric Deformable Models

Parametric deformable models provide better performance in the presence of noise than traditional edge detection methods. The greatest merit of parametric models is their computational efficiency and simplicity (Jacob et al. 2004). The merits of parametric models, however, are accompanied by several disadvantages as discussed below.

4.2.6 Disadvantages of Parametric Deformable Models

Parametric deformable models have several drawbacks associated with their operation that have generated extensive research towards solving some of those drawbacks. The disadvantages listed in this section were given by Xu et al. (2000), Jacob et al. (2004) and Brigger et al. (2000) as motivation for their proposed techniques to improve parametric deformable model performance.

- ❖ Traditional parametric deformable models have two key problems associated with them, namely, the need for the initial model to be close to the desired boundary if it is to converge satisfactorily and the failure to progress into boundary concavities.
- ❖ The parameterisation of the model is negatively affected by the use of gradient magnitude-based energy which is also parameter-dependent.
- ❖ The discrete nature of the model parameterisation results in inaccuracies in the calculation of normals of curvature.
- ❖ It is difficult to accurately and efficiently determine the weighting factors for the internal energy.
- ❖ Convergence of parametric models is slow due to a large number of coefficients that have to be optimised during evolution.
- ❖ Parametric models are prone to self-intersections which are computationally costly to prevent.

4.3 2-D Geometric Deformable Models

Geometric deformable models were independently proposed by Caselles et al. (1993) and Malladi et al. (1995) to solve the primary limitations of parametric models. They are based on curve evolution theory and implemented using the level set method. With this type of model, curves are evolved using only geometric measures and the evolution is coupled with the image data to highlight object boundaries.

4.3.1 Curve Evolution Theory

Curve evolution theory facilitates the study of curve deformation using only geometric measures such as the unit normal and the curvature as opposed to the quantities that depend on parameters such as the derivatives of an arbitrary parameterised curve.

For a dynamic curve, $\boldsymbol{X}(s,t) = [x(s,t), y(s,t)]$, the evolution of the curve along its normal direction can be characterised by:

$$\frac{\partial \boldsymbol{X}}{\partial t} = V(\kappa)N \qquad (\mathbf{4.18})$$

where $V(\kappa)$ is the speed function that determines the speed of evolution, κ is the curvature of the curve, N is the inward unit normal.

Curvature deformation and constant deformation are the most common curve deformations in curve evolution theory (Xu et al., 2000). Curvature deformation is given by;

$$\frac{\partial \boldsymbol{X}}{\partial t} = \alpha \kappa N \qquad (\mathbf{4.19})$$

where α is a positive constant. Curvature deformation has a similar effect as the use of the elastic internal force in parametric deformable models.

Constant deformation is given by;

$$\frac{\partial \boldsymbol{X}}{\partial t} = V_{\mathbf{0}} N \qquad (\mathbf{4.20})$$

where $V_{\mathbf{0}}$ is a coefficient that determines the speed and direction of deformation. Constant deformation has a similar effect as a pressure force for parametric deformable models. Curvature and/or constant deformation couples the speed of deformation with the image data such that the curve evolution stops at object boundaries.

4.3.2 The Level Set Method

The level set method suggested by Osher and Sethian (1998) for implementing curve evolution provides a basis for the numerical scheme that is used for geometric deformable models. The curve to be deformed is represented implicitly as a set of a 2-D scalar function, the level set function. The level set is defined as a set of points that have the same function value. The purpose of the level set function is to provide an implicit representation of the evolving curve. The level set method evolves a curve by updating the level set function at fixed coordinates through time. The level set function remains a valid function even if the embedded curve changes its topology (number of components and the homology of each component).

The method is summarised as follows: the contour of the object, *C*, is modelled as the zero level set, $\psi = 0$, of a Lipschitz function, ψ, such that $\psi(s, t = 0)$ is positive in a region in which *s* is inside the contour of the object and negative outside the contour of the object. $\psi(s, t = 0)$ is equal to zero on the contour of the object.

The evolution of *C* is given by the zero level set of the level set function, $\{s, t | \psi(s, t = 0)\}$. An evolving speed function is then defined in a direction normal to *C*. A technique proposed by Malladi et al. (1995) involves separating the speed function into two components, one of

which is independent of the geometry of the contour and another which is geometry-dependent. The contour is then allowed to evolve at constant speed by equating the geometry-dependent component to zero. The speed function is then expressed as a function of the image function convolved by a Gaussian filter, the maximum and minimum image gradients and an adjustable parameter. An expression is then constructed to represent the change in ψ with time and a finite difference method is applied to solve the expression iteratively. During evolution, when the curve reaches maximum image gradient the speed function becomes zero and the zero level set will then represent the contour of the target object.

4.3.3 Advantages of Geometric Deformable Models

Geometric models have more advantages than parametric models and their ability to automatically handle topological changes is the merit most often referred to in the literature. The advantages discussed in this section were identified by Xie and Mirmehdi (2004), Han et al. (2003) and Xu et al. (2000).

- Geometric models offer great flexibility since they can handle topological changes automatically without resorting to dedicated contour tracking.
- Geometric models generally have a larger capture range than parametric models.
- A geometric model can split and merge during evolution and therefore if properly initialised it can detect several objects simultaneously.
- The intrinsic geometric properties (normal vector and curvature) are easily computed using the level set function.
- Contours evolved in a geometric framework are not prone to self-intersections due to level set representation, the entropy conditions imposed during evolution and the method used for computing isocontours.

4.3.4 Disadvantages of Geometric Deformable Models

While geometric models were developed in an effort to solve the limitations associated with parametric models, methods for their formulation and operation also have some drawbacks which have driven researchers to find several alternative implementation methods. The disadvantages discussed in this section were identified by Li et al. (2005), Xie and Mirmehdi (2004), Jacob et al. (2004) and Han et al. (2003) as motivating factors for their work in which they aim to solve these problems.

- A geometric deformable model is more computationally expensive than a parametric deformable model.
- The level set function needs periodical re-initialising during evolution. Most published methods fail to re-initialise the level set function to a signed distance function if the level set function is far away from a signed distance function.
- Standard geometric models tend to leak through the boundary if an object of interest has gaps in its boundary and these models may generate shapes that have inconsistent topology with respect to the target object.
- The level set scheme used for numerical implementation makes it difficult to impose arbitrary geometric or topological constraints on the evolving contour indirectly through the higher dimensional hyper-surface.
- The traditional formulation for geometric models makes it difficult for user-defined external forces to be added.

4.4 Further Developments of Deformable Models

In light of the disadvantages of traditional deformable models discussed above, numerous methods seeking to improve their performance have been developed. A few of the published developments are discussed in the sections that follow.

4.4.1 Developments Related to Parametric Deformable Models

Most of the work involving parametric models has been directed at increasing the capture range for the snakes in order to eliminate the need to initialise a snake close to the desired border. While some researchers have recommended geometric deformable models to solve the primary difficulties of parametric deformable models, some have cited high computational complexity and other disadvantages discussed earlier as motivation for finding solutions in the realms of the traditional parametric models and B-snakes.

An external force field referred to as a gradient vector field (GVF) was developed by Xu et al. (1998) in an effort to solve the primary problems associated with parametric deformable models. The GVF is a dense vector field that is derived from an image by solving a vector diffusion equation that diffuses the gradient vectors of a gray-level image; this results in a large capture range for a deformable model that uses the GVF as its external force and allows the model to be initialised far from the desired border. The GVF differs from traditional

parametric models in that it cannot be expressed as a negative gradient of a potential energy function and unlike traditional parametric models which are formulated using energy minimisation, a deformable model using the GVF is formulated from a dynamic force equation. Also, unlike traditional parametric models, a GVF parametric model does not require prior knowledge about whether to shrink or grow during evolution.

Formulation of a GVF-based deformable model uses Equation 4.9 as a basis, replacing the external force, $\nabla P(\boldsymbol{V})$, with a novel external force field, $X(\boldsymbol{V})$, the GVF:

$$\frac{\partial \boldsymbol{V}}{\partial t} = \alpha \left|\frac{\partial \boldsymbol{V}}{\partial s}\right|^2 - \beta \left|\frac{\partial^2 \boldsymbol{V}}{\partial s^2}\right|^2 + X(\boldsymbol{V}) \qquad (\mathbf{4.21})$$

An edge map, $f(x)$, is derived from the gray-scale image to be segmented. The gradient of the edge map has vectors pointing towards edges and these vectors have large magnitude values near edges only.

The GVF is defined as the equilibrium solution to the vector diffusion equation given by Equation 4.22:

$$\frac{\partial U}{\partial t} = g(|\nabla f(x)|)\nabla^2 U - h(|\nabla f(x)|)(U - \nabla f(x)) \qquad (\mathbf{4.22})$$

where

$$U(x, 0) = \nabla f(x) \qquad (\mathbf{4.23})$$

The first term on the right side of Equation 4.22 is referred to as the smoothing term since it produces a smoothly varying vector field and the second term is called the data term which encourages the vector field, *U*, to be close to the gradient of the edge map computed from the image data. $g(|\nabla f(x)|)$ and $h(|\nabla f(x)|)$ are weighting parameters for the smoothing term and the data term respectively. A constant value, 0.2, was used for weighting the smoothing term so that smoothing occurred everywhere in the image while $h(|\nabla f(x)|)$ was allowed to become larger near strong edges by using a value given by Equation 4.24.

$$h(|\nabla f(x)|) = |\nabla f(x)|^2 \qquad (\mathbf{4.24})$$

Soon after the proposal of the GVF force, the same team improved on this external force to come up with the generalised gradient vector flow (GGVF). For the GGVF, the following values for the weighting parameters were used:

$$g(|\nabla f(x)|) = e^{\left(\frac{|\nabla f(x)|}{K}\right)^2} \qquad (\mathbf{4.25})$$

$$h(|\nabla f(x)|) = 1 - g(|\nabla f(x)|) \qquad (\mathbf{4.26})$$

where K is a constant.

The GGFV improved the convergence of the deformable models into long, thin boundary indentations while maintaining the large capture range of the GVF (Xu and Prince, 1998).

Yu and Bajaj (2002) proposed a new external force that can automatically generate initial contours using source points. Source points were defined as pixel points which, after diffusion of gradient vectors, have none of their neighbours point at them. They based their method on the GVF, claiming that the GVF of Xu et al. (1998) has some drawbacks that their own method solves. Their proposed technique, called gradient vector diffusion (GVD), can produce an initial segmentation that can automatically initialise contours. The initial segmentation produces over-segmented regions and a region merging technique has to be applied to improve the results. The difference between the GVF and GVD is that the GVF technique applies the diffusion on a Cartesian coordinate representation of gradient magnitude vectors while the GVD applies the diffusion on the magnitude and orientation of the gradient vectors. The description of the work in their paper suggests that the GVD method only improves the GVF's ability to move a deformable model into boundary concavities and to prevent the model from moving out of boundary gaps. The method does not improve the capture range of the GVF. The method performed well on medical images as well as microscopy images.

Li et al. (2005) proposed an external force that they referred to as the edge-preserving GVF (EPGVF). They based their work on the GVF of Xu et al. (1998), citing the disadvantages of geometric and traditional parametric deformable as well as a claim that the GVF cannot automatically handle topological changes as motivating factors. Their technique first computes an external force field, the EPGVF, and then the external force is segmented such that the image is divided into several regions according to the number of available objects. This technique facilitates automatic initialisation by letting an initial model for each object be the region boundary determined during the segmentation of the external force. It is claimed that the EPGVF has all the desired properties of the GVF but, unlike the GVF, it can prevent contours from passing over weak edges and it can automatically handle topological changes. To segment the external force, the EPGVF is constructed for an input image and then

represented by a graph. From the graph, connected components are found; these are the segmented regions of the external force. This method was tested on microscopy images of pollen and cells; making it a suitable option for this project.

Other significant developments in the traditional parametric model framework include the work of McInerney et al. (1995) and Ray et al. (2003). The work of McInerney et al. (1995) yielded topology adaptive snakes which they referred to as T-snakes. The T-snakes are based on an affine cell image decomposition (ACID) framework which provides T-snakes with the ability to split or merge according to the topology of the target object. Ray et al. (2003) added a boundary condition to the GGVF PDEs and used the solution to the resulting PDEs as an external force. This external force facilitates a parametric model that supports merging of several contours.

Jacob et al. (2004) proposed an edge-based internal energy for B-snakes that preserves the parameterisation of the curve. The internal energy maintains curve smoothness during evolution. Traditional B-snakes use a gradient magnitude-based internal energy that is parameter dependant. The edge-based internal energy proposed by Jacob et al. (2004) is independent of parameterisation. The gradient magnitude-based energy used by traditional B-snakes does not use the direction information of the gradient while the edge-based energy proposed by Jacob et al. (2004) uses the gradient direction in addition to the magnitude information resulting in a more robust implementation.

In an effort to extend the basic concept of B- snakes in order to improve speed and efficiency in an interactive environment where a user can control snake deformation, Brigger et al. (2000) proposed a formulation that eliminates the need for an internal energy for the snake curve. A snake formulated by their method uses node points that are placed on the snake curve instead of traditional B-spline control points. This technique also makes use of a multi-resolution method to improve the speed of the snakes. The multi-resolution method makes use of image pyramids and then applies the iteration procedure at a coarse level, on a small version of an image. Upon convergence, the solution is fed into the next pyramid level, which is finer than the previous level, as a starting point. The procedure is repeated until the finest level of the image pyramid is reached, which corresponds to the input image.

Traditional B-snake curves are defined by control points but in the method proposed by Brigger et al. (2000), the snake curves are defined by node points that correspond to knots in B-splines. While for splines of order zero or one, control points coincide with the node

points, for higher order splines, control points and node points differ significantly. Brigger et al. (2000) argue that user-interactivity would be more efficient if node points can be manipulated instead of control points. Performance evaluation tests showed similar performance of the proposed method to traditional B-snakes that make use of internal energies.

Other developments in B-snake-based segmentation include the work of Velut et al. (2007), who proposed to locally regularise a snake at a constant sampling step by using smoothing spline filtering that is controlled through a unique parameter, and the work of Precioso et al. (2003), who proposed a method to relax the rigid B-spline interpolation constraint to increase robustness of the B-snake in the presence of noise. Velut et al. (2007) claimed that their resulting algorithm is fast while Precioso et al. (2003) claimed that their technique increases the quality of segmentation without additional computational cost.

4.4.2 Developments Related to Geometric Deformable Models

Most of the developments proposed for geometric deformable models have been directed towards improving computational efficiency and preventing their leakage through weak boundaries. The original formulation for a geometric deformable model proposed by Caselles et al. (1993) is of the form:

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| \qquad (\mathbf{4.27})$$

where

$$c = \frac{1}{1 + |\nabla G_\sigma * I|} \qquad (\mathbf{4.28})$$

c is a speed function that links the deformable model to the image data, I, and slows down the snake evolution at object borders, κ is the curvature and V_0 is a constant that determines whether the model shrinks or expands during evolution. This scheme works well for objects that have good contrast (Xu et al., 2000) but causes the model to leak through gaps in some borders.

In an effort to solve the problem of leakage through weak edges, Caselles et al. (1995) proposed a new formulation for geometric deformable models using an energy minimisation approach to design a speed function that can pull back a model if it leaks through a boundary

gap. They proposed to add an extra stopping term to their original formulation so that the formulation became:

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| + \nabla c.\nabla \phi \qquad (\mathbf{4.29})$$

Siddiqi et al. (1998) claimed that the formulation given by Equation 4.29 still had leakage problems and proposed a formulation given by Equation 4.30.

$$\frac{\partial \phi}{\partial t} = \lambda(c\kappa|\nabla \phi| + \nabla c.\nabla \phi) + \left(c + \frac{1}{2}X.\nabla c\right)|\nabla \phi| \qquad (\mathbf{4.30})$$

where λ is a weighting parameter.

The term $\frac{1}{2}X.\nabla c$ provides an extra stopping capability that can prevent leakage through small boundary gaps but still cannot stop leakage through large boundary gaps (Xu et al., 2000).

Geometric deformable models suggested by Caselles et al. (1993) and Malladi et al. (1995) are constructed using a Lagrangian formulation that produces a PDE of a parameterised curve. A related Eulerian formulation from level set methods is then used to convert the PDE into an evolution PDE for a level set function (Li et al., 2005). A variational formulation is an alternative formulation method to the traditional method. A variational geometric model formulation derives the evolution PDE from a problem of minimising an energy functional defined on a level set function.

Li et al. (2005) proposed a variational formulation for geometric deformable models that eliminates the need for re-initialising the level set function to a signed distance function. Traditional geometric models have to be initialised as signed distance functions but they deviate from the signed distance function during evolution and need to be maintained as close as possible to the signed distance function to keep the evolution stable; the re-initialisation process achieves this goal. The formulation proposed by Li et al. (2005) includes an internal energy term that forces the level set function to remain close to a signed distance function and an external energy term that guides the snake towards object borders during evolution. Advantages of this method include flexible initialisation, in which the initial model does not strictly have to be a signed distance function, and the possibility of using a large time step to increase evolution speed.

Vemuri and Chen (2003) proposed a variational formulation which incorporates shape prior information and the resulting deformable model can perform joint image registration and segmentation. Another variational formulation is that of Chen and Vese (2001) which incorporates region-based information resulting in a model that allows flexible initialisation.

For some applications, the target object has a known topology. In such cases it is more sensible to segment the object in a way that yields the correct topology. Han et al. (2003) developed what they termed a topology-preserving level set method for deformable models that ensures that the final model has the same topology as the initial one; this is achieved by maintaining the topology of the digital object embedded in the implicit model.

Xu et al. (2000) adapted the GGVF of Xu and Prince (1998) for use with geometric models, which improved performance of geometric models on objects with gaps in their boundaries. The original GGVF was modified to include simple region-based information. The method only increased the capture range but the resulting deformable model could not automatically adapt to changes in topology.

Xie and Mermehdi (2004) proposed to introduce a diffused region force into the standard geometric model formulation. Their method integrates gradient flow forces and diffused region forces to yield what they termed a region-aided geometric snake (RAGS). This method seeks to solve the issues of leakage through weak edges and convergence at local maxima in noisy images associated with geometric deformable models. An initial segmentation can be used to generate the region force and the geometric model can be viewed as a refinement of the initial segmentation. The RAGS is more robust towards weak edges, has better convergence than traditional geometric models and provides better performance in the presence of noise.

4.5 Deformable Models in Medical Image Segmentation

Deformable models have gained popularity in medical image processing and they are widely used for both structural and functional image segmentation. Some of the successful applications reported in the literature are discussed in this section.

Deformable models have been successfully applied in the segmentation of MRI images as exemplified by segmentation of the corpus callossum (Jacob et al., 2004 and Brigger et al., 2000), of the white matter of the cerebrum (Han et al., 2003) and of cardiac MRI images (Ranganath, 1995).

Successful application of deformable models in the segmentation of ultrasound images has been reported: segmentation of the heart wall (Jacob et al., 2004), the foetus (Caselles, 1993), coronary arteries (Brigger et al., 2000) and the prostate (Fenster et al., 2007).

Deformable models have also been useful in the segmentation of positron emission tomography image sequences e.g. for segmentation of the brain (Brigger et al., 2000) and computed tomography images e.g. for segmentation of the stomach (Malladi et al., 1995).

Other applications of deformable models in medical image processing have been reported by McInerney and Terzopoulos (1996), Gao et al. (1998) and Chakraborthy et al. (1996).

5. METHODS: OBJECT DETECTION FOR DEFORMABLE MODEL INITIALISATION

5.1 Introduction

The segmentation task was divided into three parts namely, object detection for automatic initialisation, parametric deformable model-based segmentation and geometric deformable model-based segmentation. The flowchart shown in Figure 5.1 gives a general overview of the segmentation of a single image.

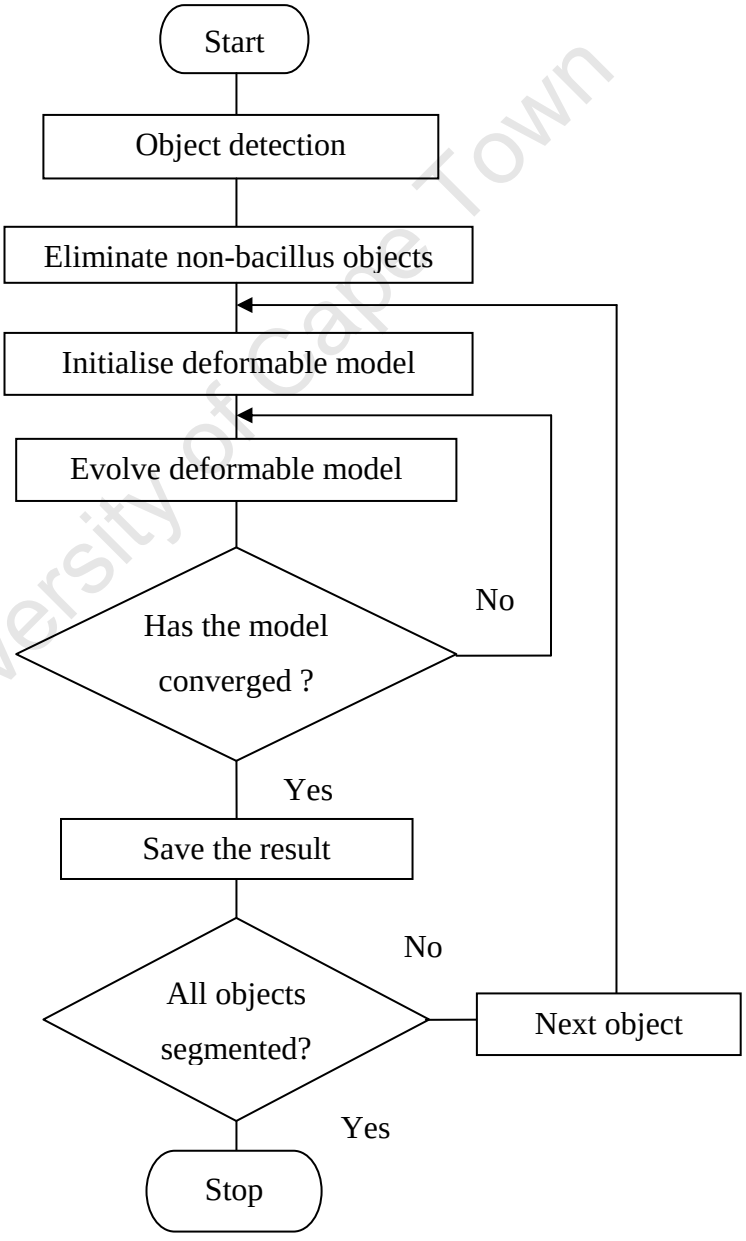


Figure 5.1: Segmentation of a single image.

Object detection serves to locate objects in an image in order to automate the initialisation of the deformable models. Objects that are not acid-fast bacilli were not subjected to deformable model-based segmentation while objects that passed the discrimination tests were subjected to deformable model-based segmentation.

The methods and algorithms that carry out the tasks described above are explained in this and the next chapter. The first three blocks of the flowchart are part of the process of automatically initialising the deformable models and are discussed in this chapter. Deformable model implementation is discussed in Chapter 6.

5.2 Initialisation

Initialisation is the process of specifying a contour or curve to represent the starting location and shape of a deformable model for evolution. The quality of initialisation has a direct effect on the result, particularly with parametric deformable model-based segmentation, in which the initial contour has to be close to the desired border. For successful segmentation, traditionally, deformable models are initialised by manually drawing a contour close to the desired border. This works well for anatomical imaging in which a single, large region of interest is required. Some sputum smear images contain hundreds of objects that are much smaller than regions of interest in medical images. This makes manual initialisation for sputum images impractical. One way of achieving automatic initialisation is to use a traditional edge detection technique to locate objects in an image and use the result as a starting point for deformable model evolution.

A number of traditional edge detection techniques were examined for this purpose, however only Canny edge detection, colour thresholding and watershed segmentation produced usable results. In some images, some objects that were not detected by the Canny edge detector were detected by the watershed segmentation and vice versa. In order to maximise the number of detected objects, the results of all three methods were combined for the initial detection of objects.

5.3 Colour Thresholding in the RGB Space

The distinct colour characteristics of ZN-stained sputum smears, which contain red bacilli against a blue background, present a useful property that can be exploited for segmentation of bacilli. Generally, good results for colour-based segmentation are obtained in the RGB space

(Gonzalez and Woods, 2002); this involves searching the entire image domain for pixels whose RGB values are within the typical colour range of bacilli.

The required colour range was specified by selecting pixels in bacillus regions; a vector of mean RGB values, obtained from bacilli in different images, of the bacillus colour was computed.

For every pixel in an image, the distance between the intensity value in each of the RGB component images and the mean RGB values was computed using the Euclidean norm, resulting in a distance vector. Components of the distance vector that were less than a threshold *T* were found and the pixels associated with these were regarded as belonging to the desired objects. The algorithm produces a binary mask with 1s in pixel locations that are associated with the required objects.

The algorithm produced very good results for images taken from samples in which all the acid-fast bacilli were stained almost uniformly with a distinct colour like the one in Figure 5.2.

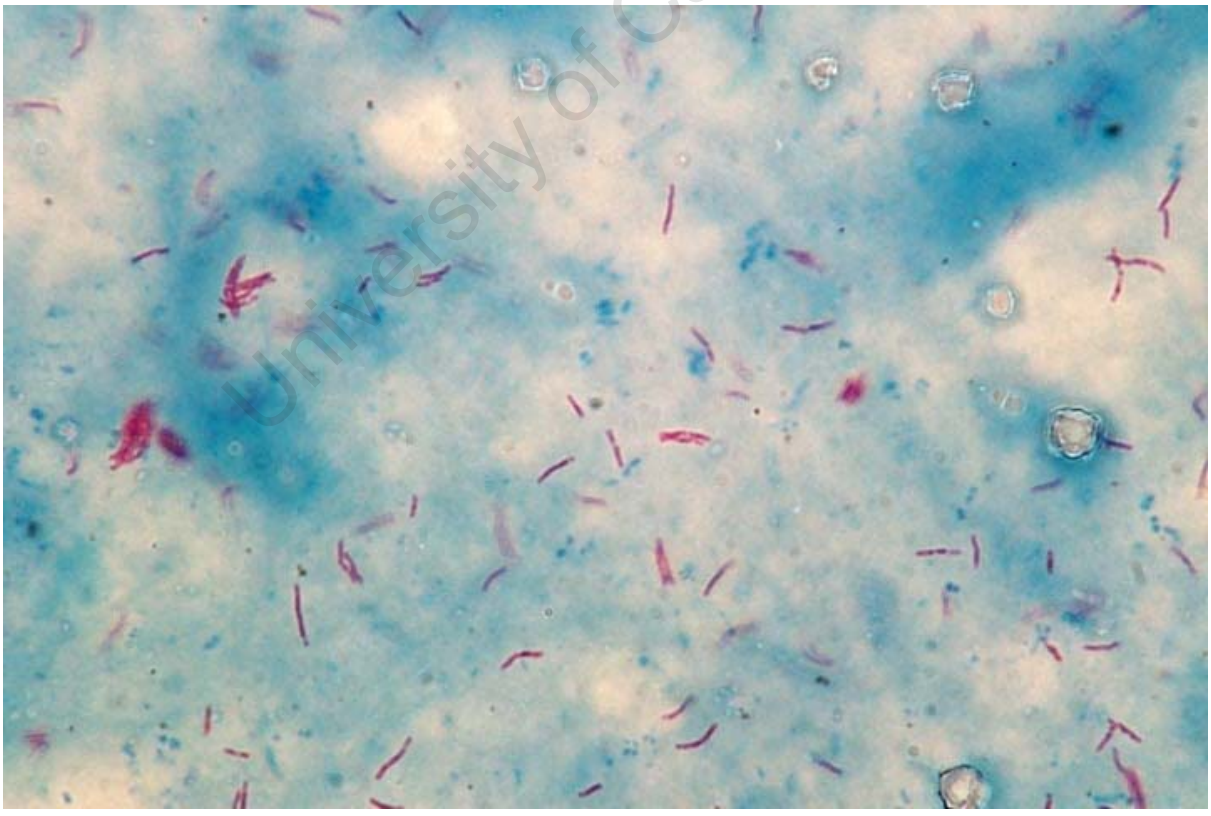


Figure 5.2: An image of a well-stained sample.

The method is sensitive to the chosen threshold. To determine the value of *T*, the standard deviation of the intensity values in each of RGB component images was computed and a value of twice the highest standard deviation of the computed values was used. Tolerance of more distant pixels increases with *T*, resulting in objects with a significantly different colour from the desired colour being segmented.

5.4 Canny edge detection

The Canny edge detector is a popular edge detection method in medical image processing due to its powerful capabilities (Najarian and Splinter, 2006); it uses image gradient magnitude and direction to find edges whereas other edge detectors use only the image gradient magnitude. The Canny edge detector also incorporates hysteresis thresholding that leads to more connected edges than other edge detectors (Ihlow and Seiffert, 2003). The steps in the application of this method are:

1. The input image is filtered using a Gaussian filter of a specified width, sigma.
2. The gradient magnitude and the direction are computed using finite difference methods.
3. Potential edge pixels are found by thresholding the gradient magnitude.
4. Two threshold values, a lower threshold and higher threshold, are applied to the result of Step 3 to detect significant edges.

The work of Ihlow and Seiffert (2003) presents a task similar to that of this project, namely the segmentation of transgenic barley cells in colour microscopy images. The Canny edge detector performed well in the hue channel of the HSI colour space. The hue channel is a periodic axis and caution has to be exercised when generalising edge detection in this channel (Ihlow and Seiffert, 2003). The authors attributed the success of the Canny edge detector in the hue channel to the fact the barley cells are not red and therefore they are not located near the red region in the hue component image. Objects of interest in the images used in this project are red in colour and therefore the method used for barley cells could not be applied directly. The Canny edge detector was, however, applied in the RGB colour space.

The characteristics of the RGB components of the sputum smear images were analysed and it was observed that the green and blue channels show great contrast between the bacilli and the background. The algorithm was therefore applied to the green and blue channels of the

images. Optimum values for the parameters were determined experimentally and the following combination gave the best performance: *sigma* = 1, *upper threshold* = 0.15 and *lower threshold* = 0.05.

5.5 Watershed Segmentation

Watershed segmentation is an edge detection technique that often produces continuous segmentation borders. This technique is particularly useful for extracting objects that are almost uniform from the background of an image (Gonzalez et al., 2004).

The watershed segmentation technique is based on the concept of catchment basins and watershed lines; the gray level image to be segmented is interpreted as a topographical surface in which the altitudes represent the intensity values. The objective of the watershed algorithm is to find the watershed lines between catchment basins. The catchment basins in the topographical image represent the regional minima of the original image and the watershed lines represent the borders of these regions. The algorithm incrementally floods the topographical surface with water from the regional minima but prevents the merging of water from different regions. At each step of the flooding process, the algorithm produces a binary image in which all OFF pixels represent pixel values that are below the flood depth (gray-level in the original image) at that step and all ON pixels represent pixel intensities that are above the flood depth. The process is repeated until water from all the regional minima almost merge; at this point the topographical image has been partitioned into catchment basins (objects) and watershed lines (borders).

The watershed algorithm is normally applied to the distance transform of a binary version of a gray-level image; however this produces an over-segmented result. A technique that uses a set of internal and external markers, can be used to improve the result of a watershed segmentation by limiting the amount of over-segmentation. A marker is a connected component in an image. An internal marker is associated with an object of interest while an external marker is associated with the background. The algorithm used here is based on the marker-controlled watershed technique described by Gonzalez et al. (2004). The algorithm is explained below.

1. The gradient image for an input image is computed from a gray-scale version of the input image using a linear filtering method.

2. Internal markers are computed by defining an internal marker as a region in which all pixels have, generally, the same intensity value and the region is surrounded by pixels of a certain minimum intensity difference, *T*, with the region.

3. The distance transform, which gives a measure of the separation of points in an image, of a binary image that contains the internal markers is computed and the external markers are found by computing the watershed regions of the distance transform image; the external markers effectively partition the image into regions that contain a single internal marker and part of the background.

4. The gradient image is modified using morphological reconstruction so that regional minima occur only at the sites of internal and external markers.

5. Finally, the watershed regions of the modified gradient image are computed, resulting in border pixels of the internal markers being identified.

The watershed algorithm was applied to the gray-scale version of an input image and to the blue component image in order to maximise the number of detected objects. The optimum value for the threshold, *T*, in step 2 was determined experimentally; a value of 20 gave good results across all the images.

5.6 Integration of Detected Objects and Elimination of Non-bacillus Objects

Background objects detected by the Canny and the watershed segmentation methods may cause undesirable snake behaviour, such as intersections, and prevent the flow of segmentation from object to object. Excluding such objects from deformable model-based segmentation avoids such errors and maintains the flow of the algorithm, and also results in a considerable reduction in the run time of the algorithm per image.

For each detected object, the size (area) was evaluated to determine the eligibility of an object for deformable model-based segmentation. The area of objects in an image is easy to compute and may be used to separate possible bacilli from non-bacillus objects. An average of 80 bacilli for each image set were manually segmented and the area of each bacillus was calculated. The mean area and standard deviation were also calculated for each image set. For each detected object, the area was computed and the object was discarded if its area was

outside the range $average \pm 2 \times standard\ deviation$ of the area values obtained for a particular set.

Objects detected by the Canny and watershed methods that were within the required area range were integrated with the results of colour thresholding into one result that contained all the segmented objects. The Canny edge detector and watershed segmentation algorithms produce binary images in which all edge pixels are represented by 1s. A MATLAB function which traces the external boundaries of objects was used to trace the detected edges in order to generate a cell array that contains an *N*-by-2 matrix, where *N* is the number of boundary pixels for the corresponding region. Each row in the matrix contains the row and column coordinates of a boundary pixel. The function returns the cell array and a label matrix for contiguous regions only (MATLAB, 2006).

To integrate the results, the row and column vectors generated for edge pixels were used to define regions of interest in the binary mask generated by the colour thresholding operation so that the binary mask contained objects detected by each of the three methods and the borders of these objects were determined.

5.6.1 Elimination by colour

The method employed here to eliminate objects based on their colour uses a similar principle to that used for colour segmentation described in Section 5.3.

In the same way that the mean bacillus area was found, the mean RGB values of the bacilli were found from 18 images captured from samples of different stain quality. The same was done to find the mean RGB values of the background. For each object that met the area criterion, the difference between the colour of the object and the average bacillus colour was found by computing the Euclidean distance between the RGB vector of each pixel and the average RGB vector for bacilli. The same procedure was done for the background and if the distance between the object colour and the bacillus colour was smaller than the object-background distance, the object would be a candidate for deformable model-based segmentation.

While the principle used for elimination based on colour is similar to that used for colour segmentation, for colour-based elimination no specific colour is being sought and hence there was no threshold to be applied. Rather, the object under examination was compared to two

colour samples and classified as either possible bacillus or non-bacillus depending on its distance from each sample.

For all the objects that meet the size and colour criteria, only objects at least 15 pixels away from the image boundary would be candidates for deformable model-based segmentation. This condition served to guard against attempts to segment objects that were partially cut off by the image boundaries. In practice, the overlap of microscope view fields or images would ensure that objects missed at the border of one image are captured in another.

6. METHODS: DEFORMABLE MODEL IMPLEMENTATION

Two main methods for implementing deformable models are reported in the literature namely, parametric and geometric methods. Both methods were explored in this project and are discussed in this chapter.

6.1 Parametric Deformable Model-based Segmentation

For the implementation of a parametric deformable model, a traditional model with a balloon force was chosen because it is computationally simple and the balloon force ensures that the snake shrinks during evolution since it is initialised outside target objects.

6.1.1 Formulation

A snake, *v(s,t)*, was formulated as a function of arc length and time. The snake, at any given time, was represented by two coordinate vectors of a series of points on the snake. The partial differential equation (PDE) used for evolving the snake is given by Equation 6.1:

$$\frac{\partial v}{\partial t} = \alpha \frac{\partial^2 v}{\partial s^2} - \beta \frac{\partial^4 v}{\partial s^4} + \kappa f_e + \lambda f_b \quad (\mathbf{6.1})$$

where *v(s,t)* is the curve function for the snake;

s is the arc length from point 1 to all the other points on the curve;

t is the time, α and β are constants that regulate the elasticity and rigidity of the snake respectively;

f_e is the external force calculated from image data. The external force stops the snake when it reaches an object border;

f_b is the balloon force that augments the external force in driving the snake towards image borders;

κ and λ are weighting parameters that determine the contribution of the external force and the balloon force respectively.

The sign of λ determines whether the snake shrinks or expands during evolution. A positive value was used in this implementation to make the snake shrink during evolution.

A snake will sometimes oscillate with a small amplitude around an object border. These oscillations can be avoided if a mechanism is incorporated into the evolution to detect convergence. The area of the region inside the curve and the length of the curve are two properties that change with evolution and these properties can be monitored to determine if the snake evolution has reached convergence. Area is the simpler quantity to compute and the area of the region inside the curve was calculated after each iteration. If the change in area in the last 10 iterations was less than a pixel, convergence would have been reached and evolution of the snake would be stopped.

6.1.2 Parameters

In order to start snake evolution, weighting parameters for the snake and the forces as well as the initial position of the snake were specified when calling the evolution function. The required parameters are discussed in this section.

For regulation of the snake, α and β were specified. These control the elasticity and rigidity of the curve respectively. Higher values for α discourage stretching while low values allow more stretching during evolution. The optimum value for α was experimentally determined to be 0.1. β regulates the curve's ability to bend during evolution, larger values enable a snake to follow a smooth border while smaller values allow the snake to follow a ragged border. The optimum value for β was experimentally determined to be 0.1.

The contribution of the external force in the evolution PDE is weighted by κ. Generally, the higher the value of κ, the sooner the snake stops evolving due to a stronger force. A universal optimum value for κ for all images was difficult to find; a good value for κ is image-dependant and it was calculated for each image as a fraction of the inverse of the external force.

The contribution of the balloon force is weighted by λ. A balloon force for parametric deformable model-based segmentation was suggested by Cohen (1991) in an effort to increase the capture range for deformable models. The initialisation technique used in this implementation ensures that the initial models are sufficiently close to the required borders so that there is no need for a large augmentation of the external force. The balloon force was included solely to ensure that the snake shrinks during deformation and this is the reason why a small value for λ was used. A value of 0.05 for λ worked well for all the images used.

The external field is given in the form of two scalar matrices, *fx* and *fy*. These two matrices are image-dependant and they are computed from the gray-level image data prior to evolution.

Snake evolution returns two vectors, *x* and *y*, whose elements are the *x* and *y* coordinates, respectively, of points on the final position of the snake. The final position of the snake represents the detected border of an object.

During deformation, it is crucial to maintain the distance between successive points on the curve at one pixel in order to maintain curve smoothness. A function analyses the snake after each iteration; it calculates the inter-point distance and computes new snake points if necessary by interpolating from the old ones.

Three conditions other than convergence necessitated the snake evolution to be broken in order to maintain stability and continuity:

1. Some objects will cause a deformable model to evolve until it virtually disappears due to lack of a border to stop the snake. Without a mechanism to guard against this condition, the algorithm fails. To avoid this condition, the minimum arc length of the evolving curve was experimentally determined. When this value is reached, evolution breaks for this object and the algorithm moves on to the next object.

2. A snake will often attempt to leave the image area during evolution, when the algorithm detects this condition, evolution is broken and the algorithm moves on to the next object.

3. The average number of iterations for true bacillus objects in low-resolution images was found to be 300 and 800 in high-resolution images. The maximum number of iterations that the algorithm allows for each object is 500 for low-resolution images and 1000 for high-resolution images. While the balloon force is given a positive weighting to force the snake to shrink, it is given a small weighting such that the external (potential) force dominates in driving the deformation of the snake. The distribution of the external force in some regions of the image will cause a snake to stray from the intended object resulting in several intersections and unusable results. A snake that has encountered self-intersections will normally undergo a lengthy evolution. The algorithm guards against this condition by limiting the number of allowable iterations.

6.1.3 Operation

This section explains the operation of the parametric deformable model-based segmentation algorithm. The flowchart shown in Figure 6.1 summarises the operation of this algorithm and an explanation follows.

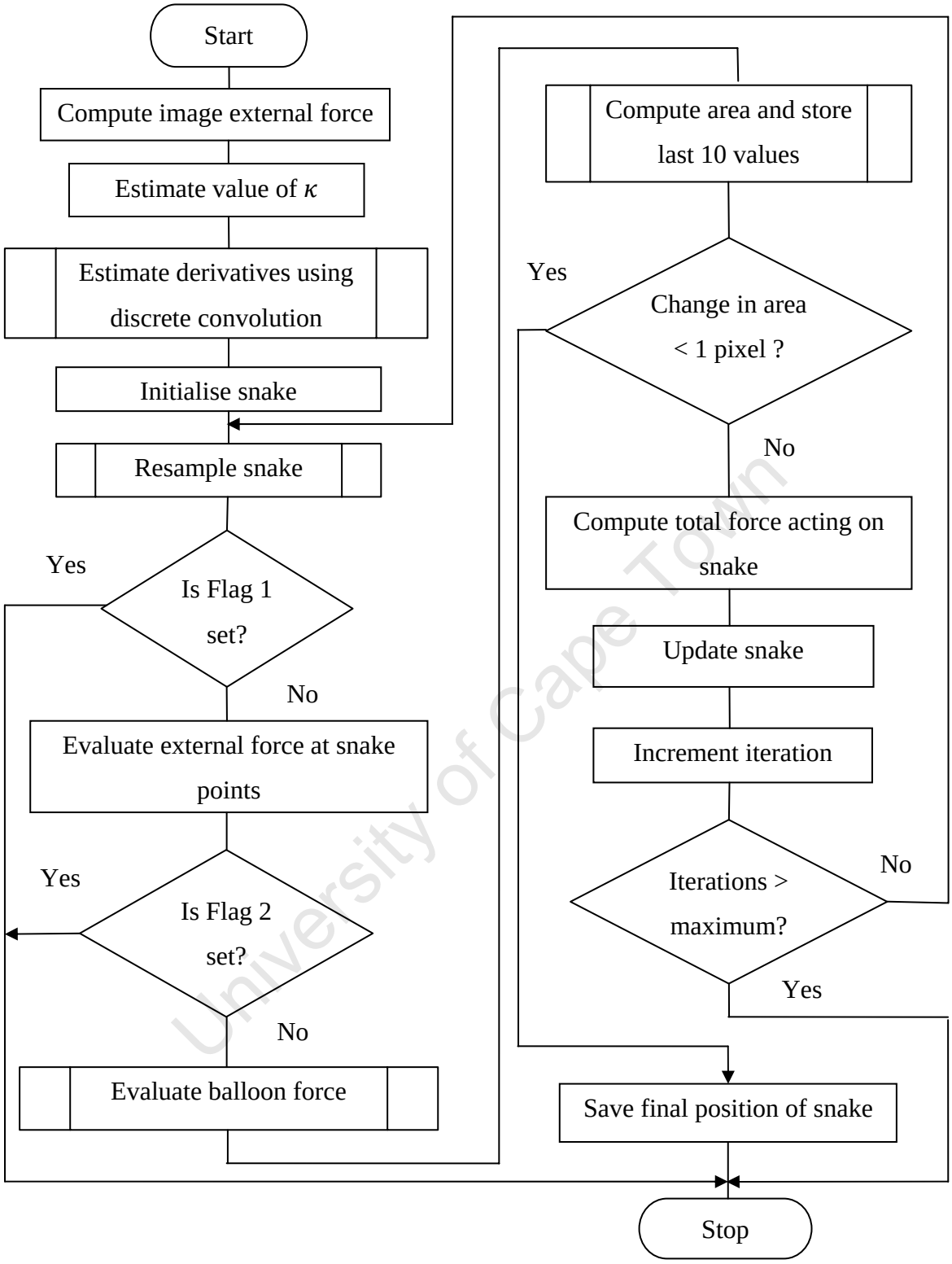


Figure 6.1: Flowchart illustrating image segmentation using a parametric model.

1. The first step in segmentation with a parametric model is to compute the external image force. The blue component image is used since it provides the best contrast between bacilli and the background. The potential energy of the image is computed using the maximum intensity value in the image; the external potential force is calculated as the gradient of the negative of the image potential energy.

2. An estimate of the external force regularisation constant, κ, is computed as the inverse of the maximum value of the external force.

3. Derivatives in the evolution equation, Equation 6.1, are approximated by discrete convolution.

4. The initial position of the deformable model is given in the form of two vectors that contain the spatial coordinates of the corners of a rectangle around the target object. The rectangle is drawn three pixels away from the extreme points of the object as illustrated in Figure 6.2; this is done automatically by using the extreme ends of the initial segmentation borders to position the corners of the rectangle. The initial snake has to be a closed contour.

5. At this stage, all the necessary parameters for snake evolution are available and the deformable model may begin deforming. The first step of each iteration, during evolution, is to ensure that the distance between successive points on the snake is uniform to maintain curve smoothness; this is achieved by a function that takes a set of points and finds the distance between point 1 and the rest of the points. The cumulative distance is calculated and if this value falls below a threshold value, the snake has become too small to continue deforming, a flag (Flag 1) is set and evolution is stopped; the threshold was experimentally determined to be 10 pixels.

   If the cumulative distance is above 10 pixels, the average distance between points is calculated; if this distance is not 1 pixel, new snake points are determined by interpolating from the old ones.

6. The external force is evaluated at each snake point by interpolation. If the external force at any point, (x,y), leads the snake out of the image area, a flag (Flag2) is set and evolution is broken off.

7. If the number of iterations is a multiple of 10, the current position of the snake is displayed on the image.

8. The balloon force is evaluated.

9. The current snake points are convolved with the approximation to the derivatives in Equation 6.1.

10. The snake is updated according to the Equation 6.1.

11. The area in the region bounded by the snake curve is computed and the latest 10 values are saved. If the change in area in the last 10 values is less than 1 pixel, the snake has converged and snake evolution is halted, returning the vectors *x* and *y* to the main function. If the change is more than 1 pixel, evolution continues by repeating steps 5 – 11 until any one of the stopping conditions is met.

12. Steps 5 through to 11 are carried out for all objects that are subject for deformable model-based segmentation.

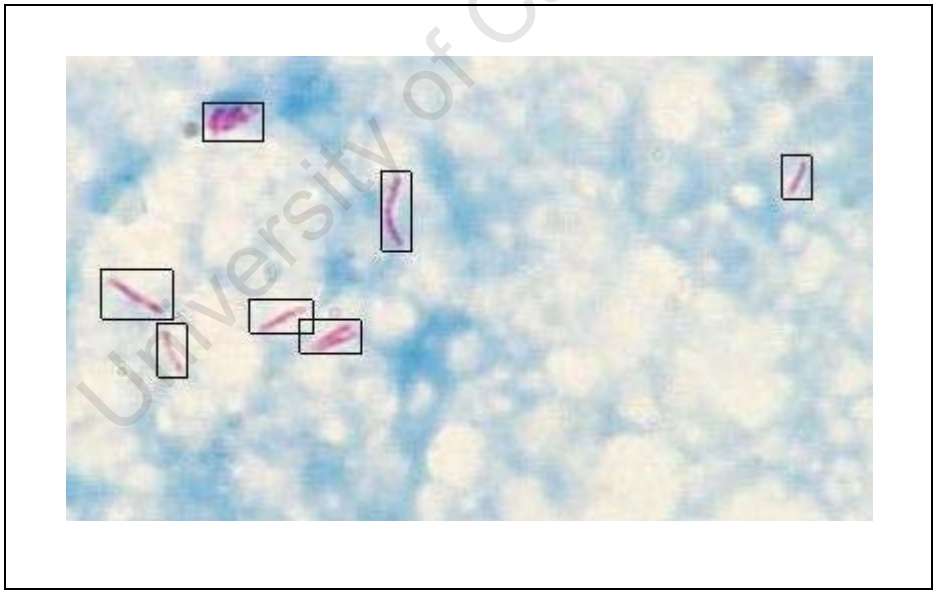


Figure 6.2: Initialisation of the deformable models.

6.2 Geometric Deformable Model-based Segmentation

The main disadvantage of geometric deformable models is computational complexity. One of the major contributors to the complexity of this type of model is the re-initialisation process

that is required to maintain the level set function close to a signed distance function (Li et al., 2005). A formulation for the contour that eliminates the need for the costly re-initialisation process would therefore reduce this complexity. Li et al. (2005) proposed a variational level set formulation for geometric deformable models that does not require the re-initialisation process and their method was adopted here.

6.2.1 Formulation

In the first step of formulation, an energy functional is defined which consists of an internal and an external energy term. The internal energy term forces the level set function to be close to a signed distance function during evolution therefore eliminating the need for re-initialisation and this improves computational efficiency of the algorithm. The external energy term guides the zero level set towards object boundaries. The level set function evolves to minimise the total energy functional, at which point the zero level set will represent the detected border.

In addition to computational efficiency, other benefits of this scheme include the use of a large time step, which increases evolution speed without compromising stability, as well as implementation through a simple finite difference scheme, which improves computational efficiency even further; the method is described below.

The internal energy term is given by Equation 6.2:

$$P(\phi) = \int_{\Omega} \frac{1}{2}(|\nabla\phi| - 1)^2 dxdy \quad (\mathbf{6.2})$$

Where ϕ is a level set function and $\mathrm{P}(\phi)$ can be considered as a metric that characterises the proximity of the level set function to a signed distance function in $\Omega \subset \Re^2$.

The variational formulation is defined as follows:

$$\mathcal{E}(\phi) = \mu \mathrm{P}(\phi) + \mathcal{E}_m(\phi) \quad (\mathbf{6.3})$$

μ is a constant that determines the weighting of the internal energy term and $\mathcal{E}_m$ is the external energy term. The evolution equation, a gradient flow that minimises the functional $\mathcal{E}(\phi)$, is given by Equation 6.4:

$$\frac{\partial\phi}{\partial t} = -\frac{\partial\mathcal{E}}{\partial\phi} \quad (\mathbf{6.4})$$

For a gray-level image, I, an edge indicator function is defined as:

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \quad (\mathbf{6.5})$$

where G_σ is a Gaussian kernel of width σ. The edge indicator function has high values at edges and marks the position of edges in an image.

The external energy for a function $\phi(x, y)$ is defined as:

$$\mathcal{E}_{g,\lambda,\alpha}(\phi) = \lambda \mathcal{L}_g(\phi) + \alpha A_g(\phi) \quad (\mathbf{6.6})$$

Where λ and α are constants, $\mathcal{L}_g$ and A_g are the weighted length and weighted area terms respectively, given by Equation 6.7 and Equation 6.8.

$$\mathcal{L}_g(\phi) = \int_\Omega g\delta(\phi)|\nabla\phi| dxdy \quad (\mathbf{6.7})$$

$$A_g(\phi) = \int_\Omega gH(-\phi) dxdy \quad (\mathbf{6.8})$$

The total energy functional given by Equation 6.3 becomes:

$$\mathcal{E}(\phi) = \mu P(\phi) + \mathcal{E}_{g,\lambda,\alpha}(\phi) \quad (\mathbf{6.9})$$

The purpose of $A_g(\phi)$ is to increase evolution speed. λ is always positive whilst α takes a positive value if the initial model is outside the desired object and negative if the initial model is inside the object.

By calculus of variations, Equation 6.9 becomes:

$$\frac{\partial \mathcal{E}}{\partial \phi} = -\mu\left[\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] - \lambda\delta(\phi) div\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) - \alpha g\delta(\phi) \quad (\mathbf{6.10})$$

The level set function, ϕ, that minimises $\frac{\partial \mathcal{E}}{\partial \phi}$ satisfies the Euler-Lagrange equation given by Equation 6.11:

$$\frac{\partial \mathcal{E}}{\partial \phi} = 0 \quad (\mathbf{6.11})$$

The steepest descent process for minimising Ɛ yields the evolution equation, Equation 6.12, for the level set function:

$$\frac{\partial \phi}{\partial t} = \mu\left[\Delta\phi - div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] + \lambda\delta(\phi)div\left(g\frac{\nabla\phi}{|\nabla\phi|}\right) + \alpha g\delta(\phi) \quad (\mathbf{6.12})$$

The terms on the right hand side of Equation 6.12 correspond to $P(\phi)$, $\lambda\mathcal{L}_g(\phi)$ and $\alpha A_g(\phi)$ respectively.

To implement the evolution equation given by Equation 6.12, the Dirac function is smoothed as a function defined by Equation 6.13:

$$\delta_\varepsilon(x) = \begin{cases} 0, & |x| > \varepsilon \\ \frac{1}{2\varepsilon}\left[1 + \cos\left(\frac{\pi x}{\varepsilon}\right)\right], & |x| \leq \varepsilon \end{cases} \quad (\mathbf{6.13})$$

Central difference is used to approximate the spatial derivatives, $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$, whilst forward difference is used to approximate $\frac{\partial\phi}{\partial t}$. By applying this difference scheme, Equation 6.12 becomes:

$$\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^{k}}{\tau} = L\left(\phi_{i,j}^{k}\right) \quad (\mathbf{6.14})$$

Equation 6.14 can be re-written as:

$$\phi_{i,j}^{k+1} = \phi_{i,j}^{k} + \tau L\left(\phi_{i,j}^{k}\right) \quad (\mathbf{6.15})$$

Equation 6.15 is the iteration equation for level set evolution. The flowchart shown in Figure 6.3 summarises the stages in the geometric model-based segmentation of a single image.

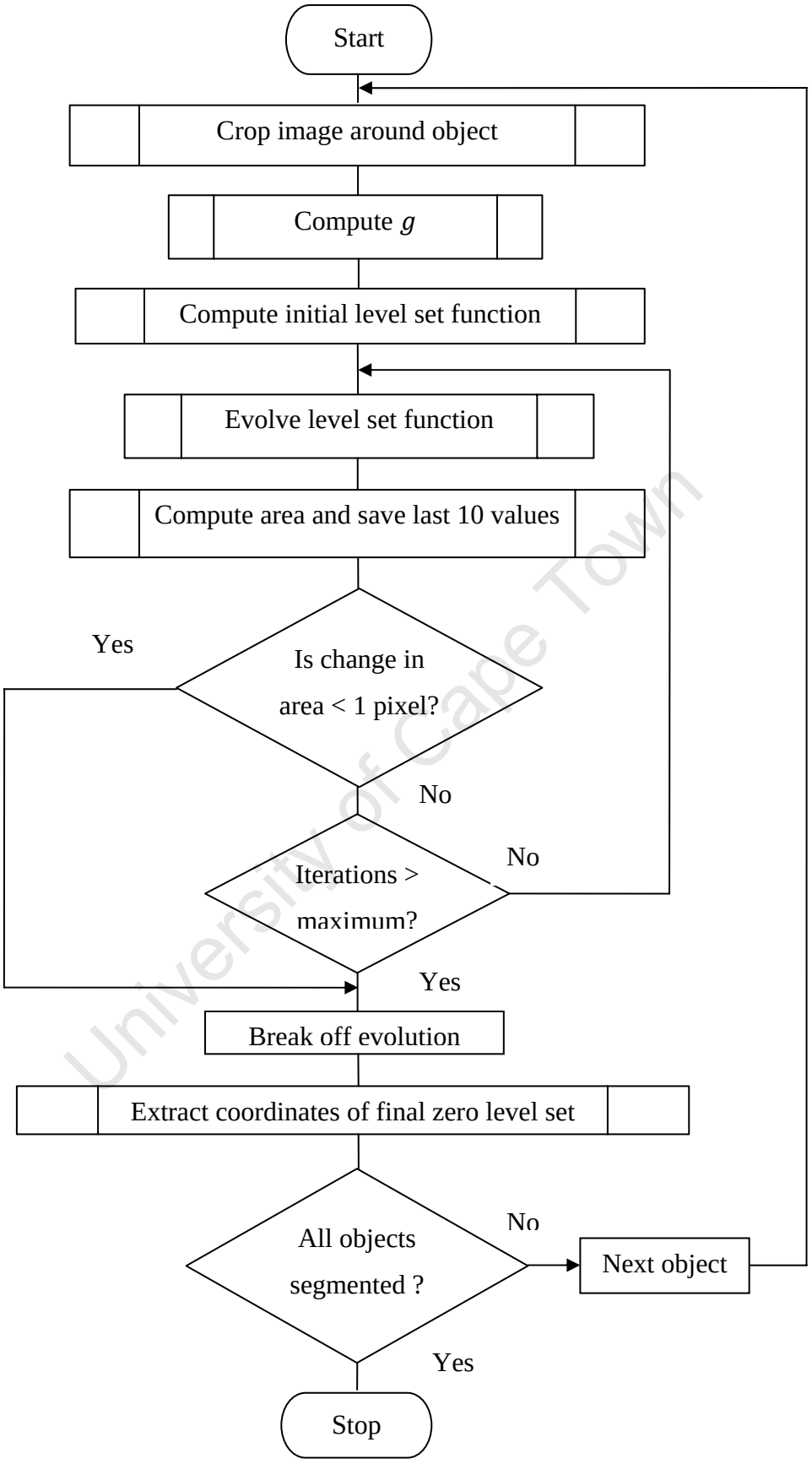


Figure 6.3: Flowchart for geometric deformable model-based segmentation.

6.2.2 Operation

To reduce the amount of information processed during evolution of the model, for each object, the image is cropped 10 pixels from the initial snake; this reduces the amount of memory required to run the algorithm and also increases the processing speed. The algorithm can be divided into three main stages namely calculation of the edge indicator function, initialisation and evolution.

i) Edge indicator function

The first step is to calculate the edge indicator function, g; this is carried out as follows:

1. The input colour image is separated into its RGB component images and the blue component image is used since it provides the best contrast between bacilli and the background.

2. The gray-level image is smoothed by Gaussian convolution.

3. The gradient, f, of the smoothed image is computed and the magnitude of the gradient at each pixel is used to compute the edge indicator function:

$$g = \frac{1}{1+f} \quad (\mathbf{6.16})$$

ii) Initialisation

While traditional level set methods require the level set function to be initialised as a signed distance function, the method applied here allows flexible initialisation in which the initial level set function does not strictly have to be a signed distance function. For each object, the initial level set function is calculated as follows:

1. A region of interest is selected by drawing a rectangle around an object in the same way as discussed for the parametric model. This method is referred to as region-based initialisation of the level set function. The advantage of region-based initialisation is that the region of interest can be automatically obtained by thresholding and the level set will evolve with its zero level set converging accurately on the border in the region of interest; an advantage that has been fully exploited here.

2. A constant value, *C*, is used to define a binary level set function and the initial level set is calculated as follows:

$$u_0 = C \times 2(0.5 - BW) \quad (\mathbf{6.17})$$

where *BW* is a binary image with 1s in the position of the region of the smoothed image defined by the selection rectangle. The optimum value for *C* was experimentally determined as 8. At this point all the necessary parameters for evolution are available and evolution of the level set function may begin.

iii) Evolution of the level set function

The function that evolves the level set function seems rather simple for the complex computations that it carries out;

1. Firstly, the level set function is sampled and adjusted to ensure that it satisfies the Neumann boundary condition. This means that the solution to the evolution partial differential equation, Equation 6.12, will satisfy the Neumann boundary condition, which specifies the values that the derivative of a solution to the evolution partial differential equation has to take on the level set function; this ensures that the evolution of the deformable model remains stable.

2. The gradient of the level set function is computed and normalised to give the *x* and *y* components *Nx* and *Ny* respectively.

3. The Dirac function in Equation 6.13 is evaluated using the current level set function. The optimum value for ε, the parameter in the definition of smooth Dirac function in Equation 6.13, was experimentally determined as 1.5.

4. The curvature κ is computed as a sum of the *x* and *y* components of the gradients of *Nx* and *Ny* respectively.

5. The weighted length term is computed as follows:

$$Wl = \lambda \times diracU\big((vx \times Nx) + (vy \times Ny) + (g \times \kappa)\big) \quad (\mathbf{6.18})$$

   where *vx* and *vy* are the *x* and *y* components of the gradient of the edge indicator function, respectively, *diracU* is the Dirac function computed in step 3.

6. A finite difference approximation, *l*, of Laplace's differential operator applied to the current level set function, u_n, is computed:

$$l = \frac{1}{4}(\nabla^2 u_n) \quad (\mathbf{6.19})$$

7. The penalising term is modified using l as follows:

$$Pterm = \mu \times (4l - \kappa) \quad (\mathbf{6.20})$$

8. The weighted area term is calculated as follows:

$$Wa = \alpha \times diracU \times g \quad (\mathbf{6.21})$$

9. The level set function is updated through a time step, δ, as follows:

$$u_{n+1} = u_n + \delta(Wl + Wa + Pterm) \quad (\mathbf{6.22})$$

10. The area of the region bounded by the contour is calculated and if the change in the latest 10 values is less than a pixel then the evolution has converged and snake evolution is stopped. If the change in area is more than a pixel, evolution repeats until convergence is reached or until the maximum number of iterations is reached; a maximum of 450 iterations is allowed for low-resolution images and 900 for high-resolution images.

For each object, the evolution function returns the final level set function. The zero level set of the final level set function represents the detected object border. The *x* and *y* coordinates of points on the zero level set are extracted and saved.

6.3 Validation

The need for evaluating segmentation algorithms arises in cases where two or more methods seem to give similar results, visually, or for a single algorithm, to fine-tune parameters to obtain the best possible result. The primary reason for evaluating the segmentation schemes for this project was to determine the segmentation scheme that gives better results.

Two key issues are generally addressed when validating segmentation algorithms:

- What is the true result that will be used for comparison?
- What quantity should be measured and how?

There are two ways in which the above issues may be addressed, namely supervised and unsupervised validation (Sonka et al., 2008).

Unsupervised validation methods make use of assumptions about good segmentation such as:

- A good segmentation produces objects with high intra-region uniformity.
- An image is segmented into homogeneous regions of simple shapes.
- Segmented regions have high contrast.

Unsupervised validation methods use global image statistics derived from all pixels in an image or regional statistics of segmented regions to verify the assumptions stated above (Sonka et al., 2008).

In supervised validation methods, the true result is assumed to be available. The true result is normally obtained through establishing a ground truth by manually drawing a contour to represent the true border. Two common supervised validation approaches assess mutual overlap and border positioning errors (Sonka et al., 2008).

The mutual overlap approach computes the area of overlap between the ground truth and automatically segmented region. The mutual overlap metric is given by:

$$M_{MO} = \frac{2MO}{A_1 + A_2} \qquad (\mathbf{6.23})$$

where A_1 is the area of the automatically segmented region, A_2 is the area of the manually segmented region and MO is the area of overlap (Bowyer, 2000).

This metric often fails to fully portray the qualitative difference between two different segmentation techniques. It performs well when the distances from a segmented boundary to the ground truth are distributed unimodally with low variance (Sonka et al., 2008).

A supervised approach was taken in validating the segmentation algorithms in this project by determining positioning errors between automatically detected borders and manually determined borders. Border positioning errors are a common validation approach in the literature and the metric used for quantifying differences between contours in this project is the Hausdorff distance.

6.3.1 The Hausdorff Distance Metric

The Hausdorff distance measures the extent to which each point of a model set lies near some point of an image set and vice versa. This distance can be used to determine the degree of

resemblance between two objects superposed on one another (Huttenlocher et al., 1993). The Hausdorff distance can be defined as follows:

Given two finite set points $A = \{a_1, a_2, \dots a_p\}$ and $B = \{b_1, b_2, \dots b_q\}$, the Hausdorff distance between the two sets is:

$$H(A,B) = \max\big(h(A,B), h(B,A)\big) \quad (\mathbf{6.24})$$

where $h(A,B)$ is the directed Hausdorff distance from *A* to *B* and $h(B,A)$ is the directed Hausdorff distance from *B* to *A* and defined as:

$$h(A,B) = max_{a\in A} min_{b\in B} \|a-b\| \quad (\mathbf{6.25})$$

$$h(B,A) = max_{b\in B} min_{a\in A} \|b-a\| \quad (\mathbf{6.26})$$

$\|a-b\|$ is a distance metric; the metric used in this project is the Euclidean distance. The directed Hausdorff distance is asymmetric; usually the directed Hausdorff distance from *A* to *B* is not equal to the directed Hausdorff distance from *B* to *A*. Effectively, the Hausdorff distance is computed by finding the minimum distance between each point of a set and all the points of a model set. The maximum value of these distances is the directed Hausdorff distance from the set to the model. The directed Hausdorff distance from the model to the set is similarly computed; the larger of the two directed Hausdorff distances is the Hausdorff distance between the set and the model.

The Hausdorff distance can be used to evaluate segmentation results by letting *A* in Equation 6.24 be the set of boundary pixels specified by the ground truth and *B* the set of boundary pixels determined by the segmentation technique to be evaluated.

6.3.2 Validation Procedure

To validate the accuracy of the algorithms in detecting object borders, a ground truth was established. In order to simplify the validation process, only isolated bacilli were selected from the detected objects and manually segmented; clustered bacilli were not included in the validation process. Manual segmentation was carried out under supervision and guidance of a pathologist who verified the accuracy of the manually drawn contours.

The Hausdorff distance calculated is given in pixels; a small Hausdorff distance indicates a close match in the automatic and manual borders.

7. RESULTS AND DISCUSSION

7.1 Introduction

After all the algorithms were developed, extensive tests were carried out to assess and evaluate performance. The algorithms were applied to all images in the image sets given in Section 1.3.

Conventional microscopy for TB typically uses a 100-x objective lens (Steingart et al., 2006), however, it is anticipated that the microscope under development at the MRC/UCT Medical Imaging Research Unit will, in its final state, have a 40-x objective lens that enables larger fields to be viewed and hence faster slide processing. Images captured at 40-x and 100-x magnification were included in this project to compare the success of the algorithms when applied to images captured at these magnifications. Figure 7.1 shows images captured at different magnification and image resolution.

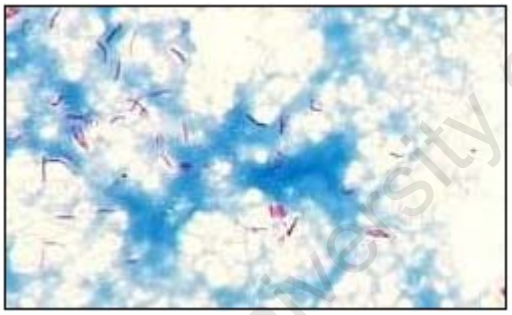
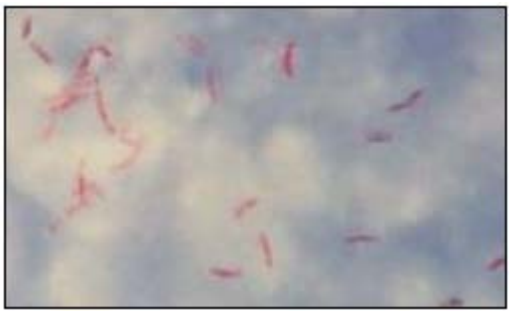
Images captured at 40-x	**Images captured at 100-x**
 480x720 pixels	 480x720 pixels
 1200x1792 pixels	 1200x1792 pixels

Figure 7.1: Bacillus size at different magnification and resolution.

The images were captured from slides of different stain quality ranging from well-stained samples, in which the bacilli and the background have stained almost uniformly and the bacilli are distinct from the background, to badly-stained samples in which the bacilli are not distinct from the background. The algorithms used in this project produced good results on well-stained, less dense images while performance was reduced on badly-stained images with high bacillus density. Images with clumped bacilli such as those shown in Figure 7.2 were not considered in this project.

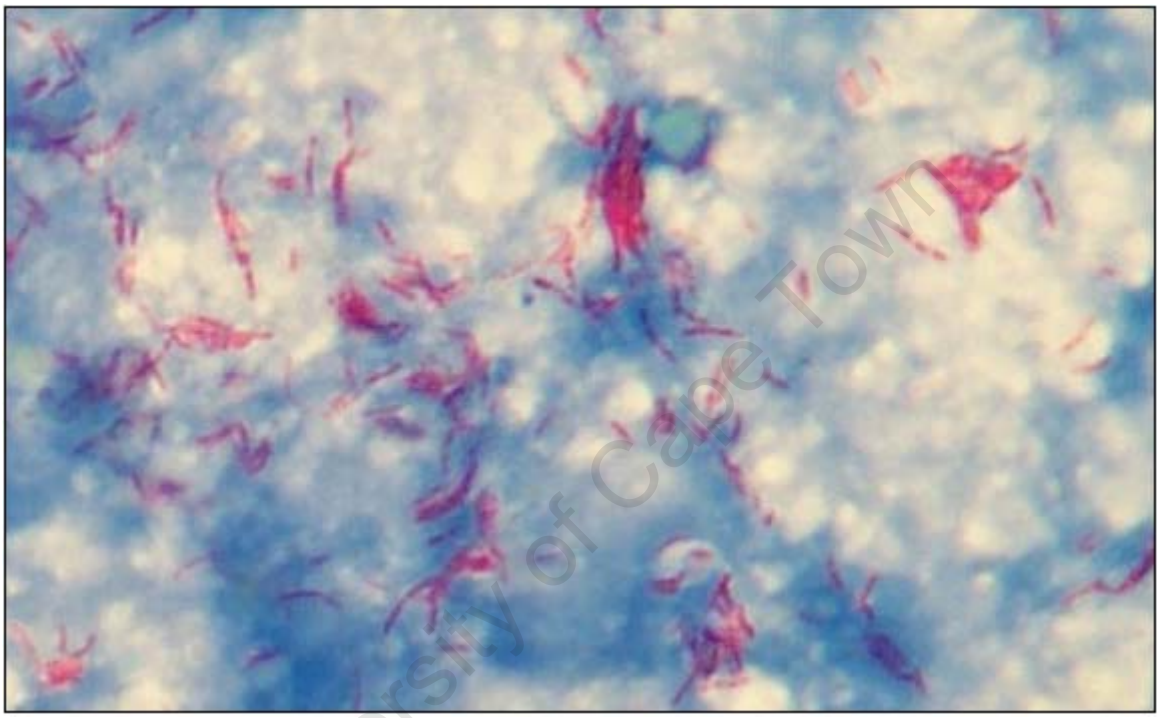


Figure 7.2: Typical image with clumped bacilli.

This chapter is divided into sections that discuss the results of initialisation and the results of parametric and geometric deformable model-based segmentation.

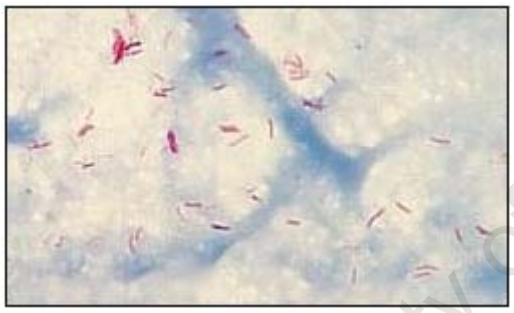
7.2 Initialisation Method

Colour segmentation in the RGB colour space, Canny edge detection and watershed segmentation were applied to the images to locate objects in the images. The results of all the techniques were integrated to maximise the number of detected objects. The methods could not be used in isolation since, for each method, the number of detected objects is limited.

7.2.1 Colour segmentation

In order to maximise the number of objects that may be detected, the colour segmentation algorithm used in this implementation uses a threshold value of 55, two times the highest standard deviation of the RGB values of bacilli, which determines the acceptable colour range around the mean bacillus colour. The colour thresholding algorithm produces a binary image with 1s in the pixel locations that are within the required colour range; the borders of the resulting regions are determined and these are shown on the example images.

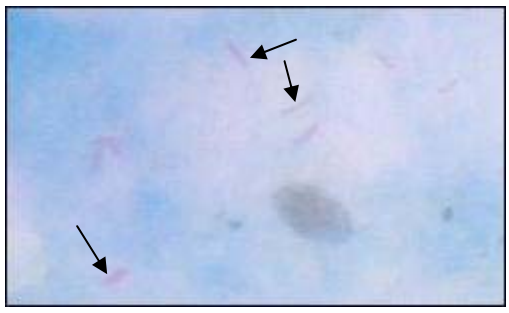
Colour segmentation works well for well-stained images in which the bacilli stain almost uniformly as illustrated by Figure 7.3, where detected objects in the typical image shown in Figure 7.3 (a) are outlined in Figure 7.3 (b). Figure 7.4 illustrates that, despite bacilli being visible in an image of a badly-stained smear, no objects are detected.



(a) A typical image captured at 40-x

(b) Outlines of detected objects

Figure 7.3: Colour segmentation of a typical image of a well-stained smear.



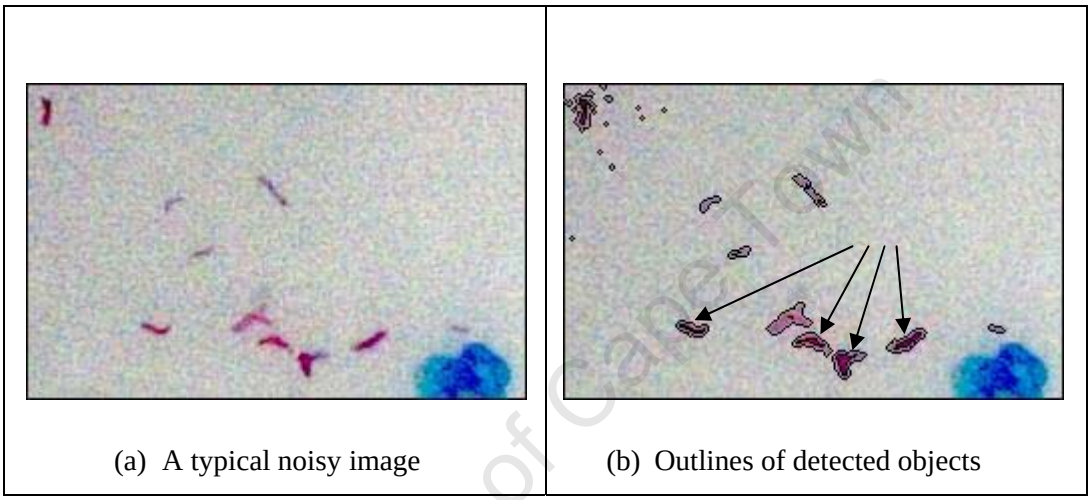
(a) A typical badly-stained smear

(b) No objects detected for bad stain

Figure 7.4: Colour segmentation of a typical image of a badly-stained smear; arrows indicate visible bacilli.

The performance of this method is independent of image size and it performs consistently on 40-x and 100-x images. The processing speed is, however, reduced for high-resolution images due to an increase in the number of pixels that have to be processed.

The method performs well in the presence of noise as shown in Figure 7.5. Double contours are due to uneven staining of some bacilli; this occurs when some pixels within a bacillus are out of the required colour range. However, during the integration of results of the initial detection methods, the regions defined by the inner contours are engulfed by the regions defined by the outer contours to form one region.



(a) A typical noisy image (b) Outlines of detected objects

Figure 7.5: Colour segmentation of a noisy image; arrows indicate objects with double contours.

Figure 7.6 shows typical results of colour segmentation obtained for bacilli at different magnification and different image resolution.

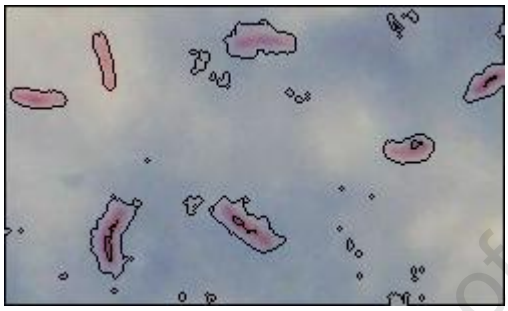
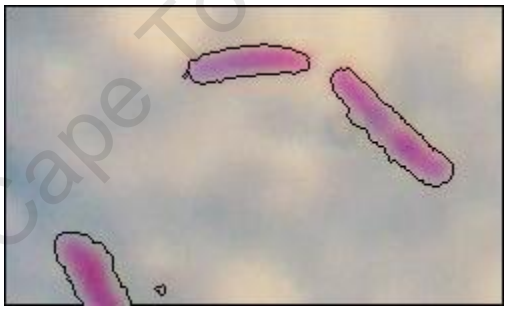
Typical images captured at 40-x	Typical images captured at 100-x
 (a) 480 x 720 pixels	 (b) 480 x 720 pixels
 (c) 1200 x 1792 pixels	 (d) 1200 x 1792 pixels

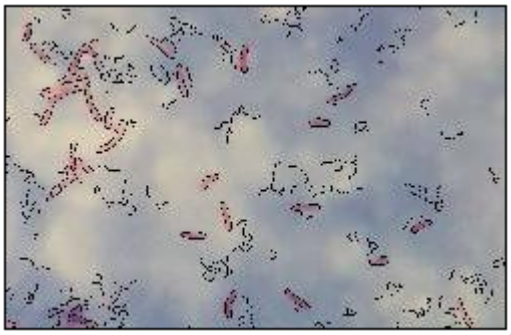
Figure 7.6: Colour segmentation of bacilli at different magnification and resolution.

7.2.2 Canny Edge Detection

Edges detected by the Canny edge detection algorithm in images with 1200x1792 pixel resolution have gaps as illustrated by Figure 7.7. The broken contours that were obtained in high-resolution images were the motivation behind the exploration of deformable models as an alternative method for segmenting images of sputum smears. It was discovered, however, that for low-resolution images (480x720 pixels), the same algorithm produces complete contours for detected objects. The algorithm produced good results only in images of well-stained samples.

(a) A typical image captured at 40-x

(b) A typical image at 100-x



(c) Canny edges

(d) Canny edges

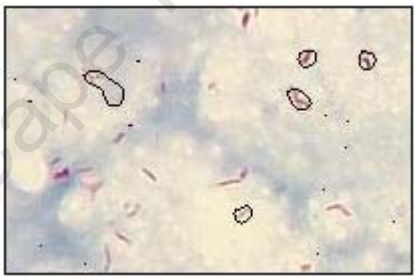
Figure 7.7: Canny edges for high-resolution images.

The optimum parameter values for Canny edge detection that produced the best results for the image sets used are *sigma* = 1, *lower threshold* = 0.05 and *upper threshold* = 0.15. This combination of parameters enabled the Canny edge detector to detect the highest number of objects; raising the value of *sigma* or any of the thresholds reduced the total number of objects detected. Experiments also showed that when the threshold values were maintained at constant values while *sigma* was increased, the number of detected objects was reduced and the algorithm captured the shapes of the bacilli with less accuracy. Virtually no detection occured at values of *sigma* higher than 6. When the value of *sigma* was maintained at 1 and the threshold values were raised, accuracy in border delineation was maintained but the number of detected objects was reduced. Reducing parameter values from their optimum slowed down the detection but had no observable effect on accuracy and number of detected

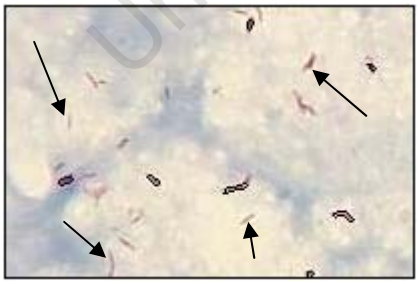
objects. The effect of different parameters on the performance of the Canny edge detector is illustrated in Figure 7.8. The outlined objects in Figure 7.8 (b), (c) and (d) are objects that the Canny edge detector would contribute during the integration of the results of the three methods by the process described in Section 5.6 at the different parameters indicated.

(a) Example image captured at 40-x (480x720 pixels)

(b) Lower threshold = 0.05, upper threshold = 0.15, Sigma = 1



(c) Lower threshold = 0.05, upper threshold = 0.15, Sigma = 3



(d) Lower threshold = 0.1, upper threshold = 0.3, Sigma = 1

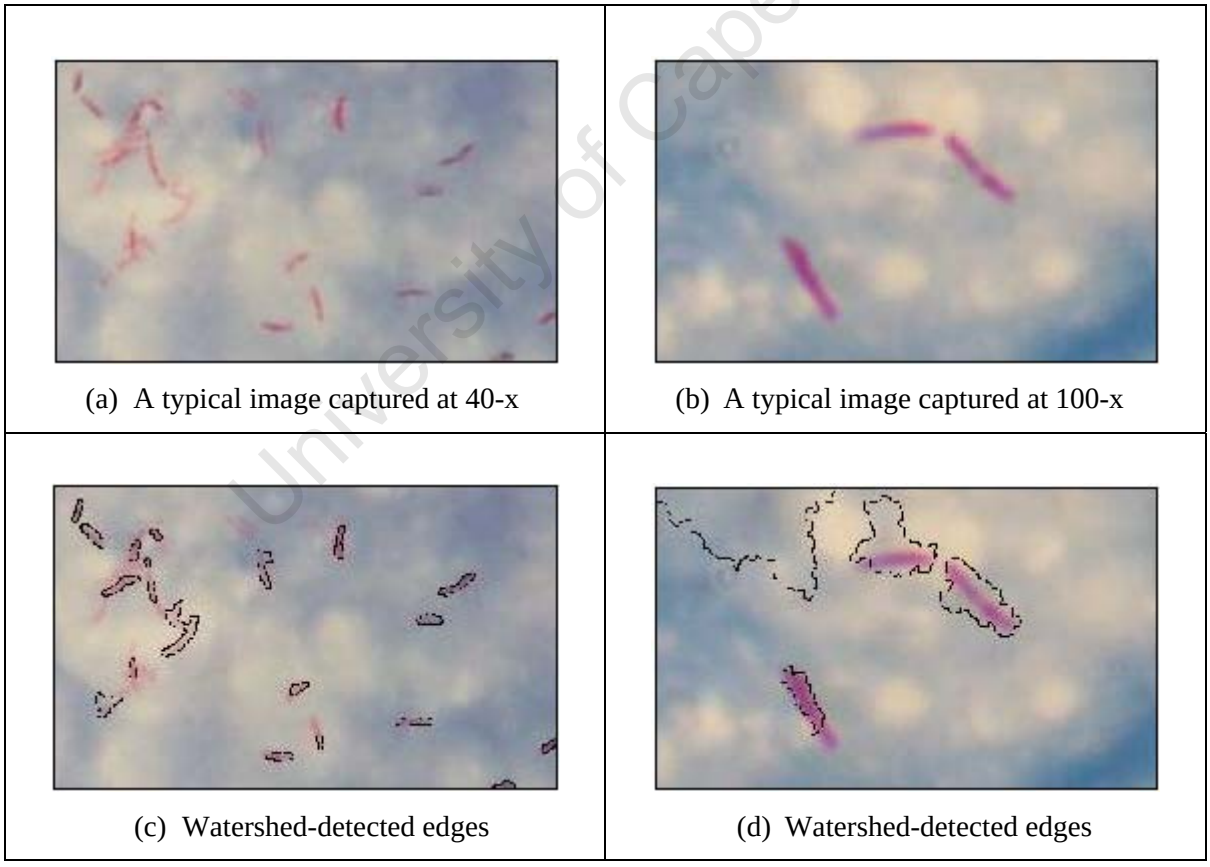
(e) Lower threshold = 0.05, upper threshold = 0.15, Sigma = 6

Figure 7.8: Typical results of Canny edge detection for different parameters; arrows indicate bacilli missed at higher threshold values.

7.2.3 Watershed Segmentation

There is only one parameter that characterises the performance of the watershed algorithm, namely the difference in gray levels of a marked area and its surrounding regions; a value of 20 enabled the algorithm to detect the highest number of bacilli in the images. A consequence of using this value was that it caused fragmentation of detected bacilli. This, however, caused no serious problems since most of the fragmented bacilli were joined during integration of the results of all the methods. A value of 40 reduced fragmentation at the cost of reducing the number of detected objects while values that are less than 20 increased fragmentation.

The watershed segmentation algorithm, like the Canny edge detector, produced broken contours for detected objects in high-resolution images as illustrated in Figure 7.9, while contours for low-resolution images were complete as shown in Figure 7.10. For both methods, the gaps in the detected edges may be attributed to noise in the higher resolution images.



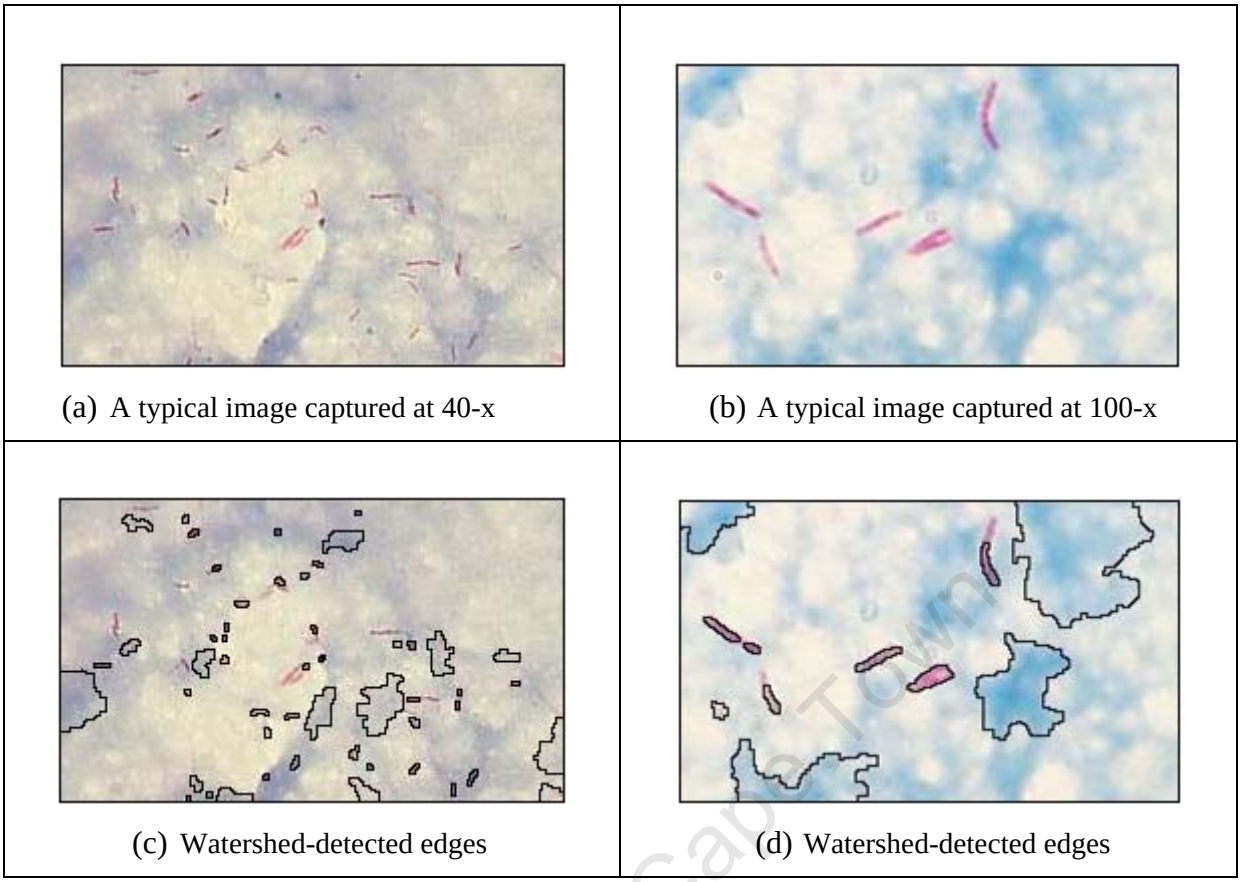
(a) A typical image captured at 40-x

(b) A typical image captured at 100-x

(c) Watershed-detected edges

(d) Watershed-detected edges

Figure 7.9: Typical watershed-detected edges in high-resolution images.



(a) A typical image captured at 40-x

(b) A typical image captured at 100-x

(c) Watershed-detected edges

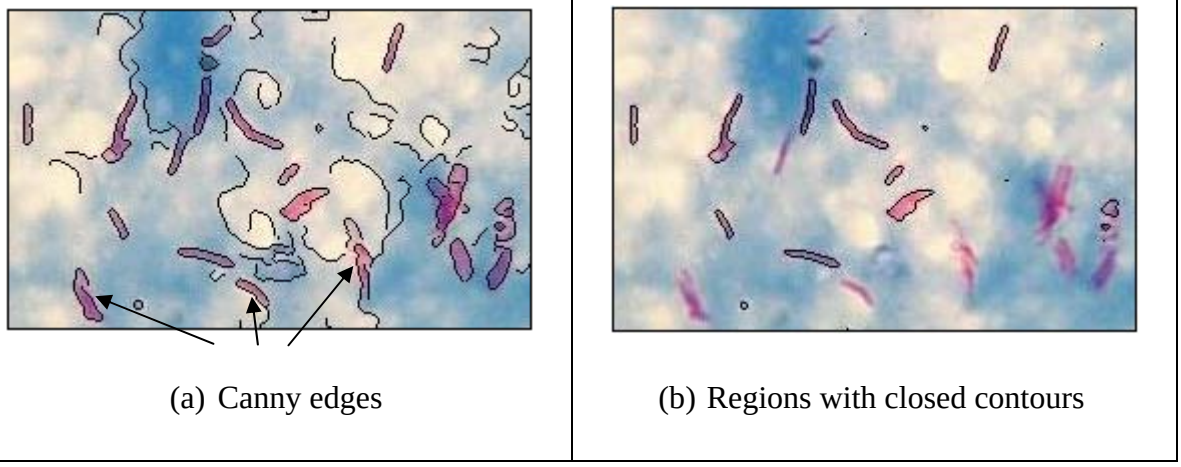
(d) Watershed-detected edges

Figure 7.10: Typical watershed-detected edges in low-resolution images.

7.2.4 Combined Object Detection

Results indicated that each of the three methods used for object detection cannot always detect all objects in an image and that some objects that cannot be detected by one method can be detected by the other two methods. A combined result therefore has more detected objects than a result obtained by applying one method.

Contours with unconnected ends are discarded during the integration of results by the process described in Section 5.6. Figure 7.11 illustrates how contours with unconnected ends result in the loss of bacilli; such contours are more common in the results of the Canny edge detector.



(a) Canny edges

(b) Regions with closed contours

Figure 7.11: The loss of bacillus objects caused by unconnected Canny edges in a typical low-resolution, 100-x image; arrows point to bacillus objects lost due to unconnected edges.

In Figure 7.11 (a), all present bacilli have contours associated with them; the method described in Section 5.6, used for integrating the results of initial segmentation, discards contours with unconnected ends because such contours do not define contiguous regions.

Figure 7.12 shows typical objects detected by the Canny edge detector in high-resolution images that are recognised by the algorithm during the integration of the results of initial segmentation.

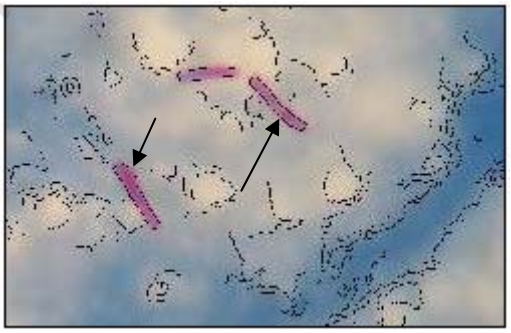
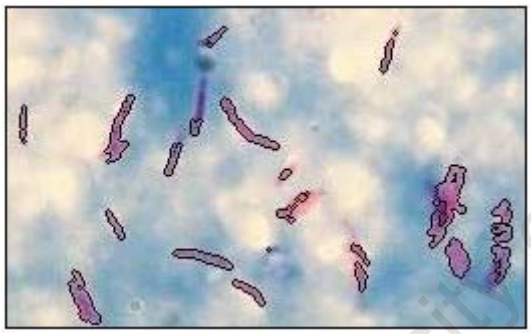


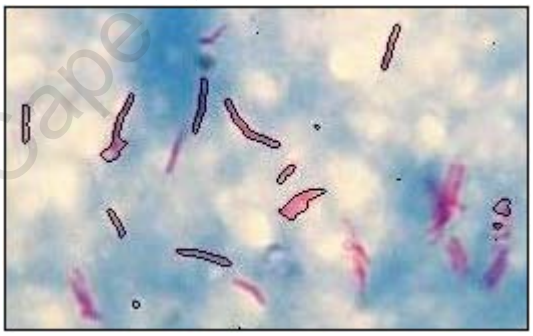
Figure 7.12: Canny edges for a typical 100-x, high-resolution image; arrows indicate objects recognised during integration of results.

Figure 7.13 illustrates how the number of objects detected in a typical 100-x, low-resolution image is optimised by integrating the results of the colour thresholding, Canny edge detection and watershed edge detection in images of well-stained smears.

(a) Example image

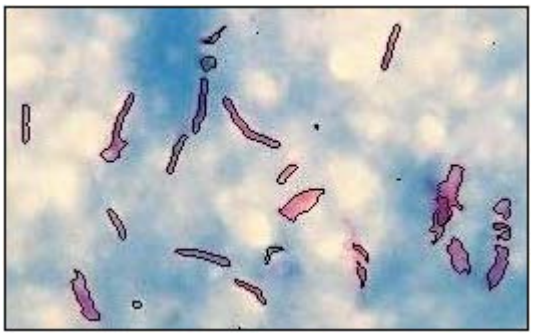


(b) Results of colour segmentation



(c) Results of Canny edge detection

(d) Results of watershed segmentation



(e) Integration of all results

Figure 7.13: Integration of initial segmentation results.

The objects outlined in Figure 7.13 (c) and (d) indicate objects detected by the Canny edge detector and the watershed algorithm respectively, with contours that have connected ends; contours with unconnected ends have been discarded.

Table 7.1 summarises the results of combining the results of the three methods in 10 images of each image set; only images of well-stained smears were considered.

Table 7.1 Detection performance of the combined initialisation method

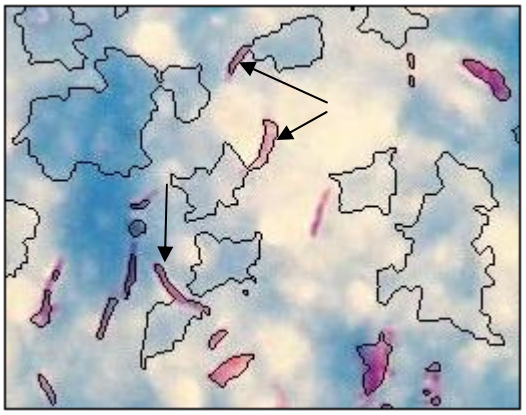
Image set		Bacilli present	Undetected bacilli	% detection
Magnification	Resolution (pixels)			
40-x	480 x 720	779	14	98.2
	1200 x 1792	399	4	99.0
100-x	480 x 720	281	1	99.9
	1200 x 1792	150	20	86.7

7.2.5 Elimination of Non-bacillus Objects

In order to reduce computation time, objects whose area values were not within twice the standard deviation of the average area of bacilli were eliminated during the integration of the results of initial segmentation. Objects whose area values were within the desired range but had an average colour that was closer to the average background colour were eliminated prior to deformable model-based segmentation.

In some cases, possible bacilli were eliminated because they were joined to large objects during detection as illustrated in Figure 7.14. For slides with many bacilli, the compromise of losing a few possible bacilli is acceptable, considering the complexities brought about by extremely large objects, such as deformable model intersections and reduction in speed. Such slides would contain sufficient bacilli for a positive TB diagnosis.

(a) Example image



(b) Watershed-detected edges

Figure 7.14: Bacillus joined to large objects during edge detection; arrows indicate bacilli joint to large non-bacillus objects.

Elimination by colour worked well across all the images. This method was also successful in respect of eliminating background objects only and rarely eliminated definite bacilli; this is largely due to the distinct colour characteristics of the background and bacilli in images of ZN-stained smears. Table 7.2 summarises the performance of the elimination method in 10 images of each image set.

Table 7.2: Performance of the elimination method

Image set		**Bacilli detected**	**Bacilli eliminated**	**Non-bacillus objects eliminated**
Magnification	**Resolution (pixels)**			
40-x	480 x 720	765	25	5 374
	1200 x 1792	395	32	13 379
100-x	480 x 720	281	12	4 419
	1200 x 1792	130	5	42 389

The large numbers of non-bacillus objects detected in each image set justify the need to eliminate such objects prior to deformable model-based segmentation in order to reduce

processing time. Eliminated bacilli were due to the effect illustrated in Figure 7.14 and missed bacilli were due to images of thick smears in which the bacilli were deep in the background and therefore the characteristic reddish-pink stain was not clearly distinguishable from the background; an example of such an image is shown in Figure 7.15.

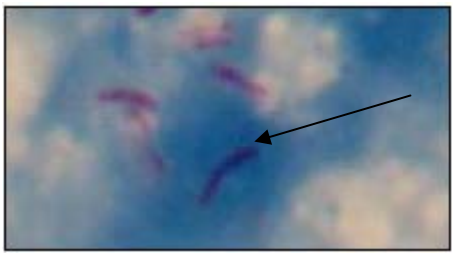
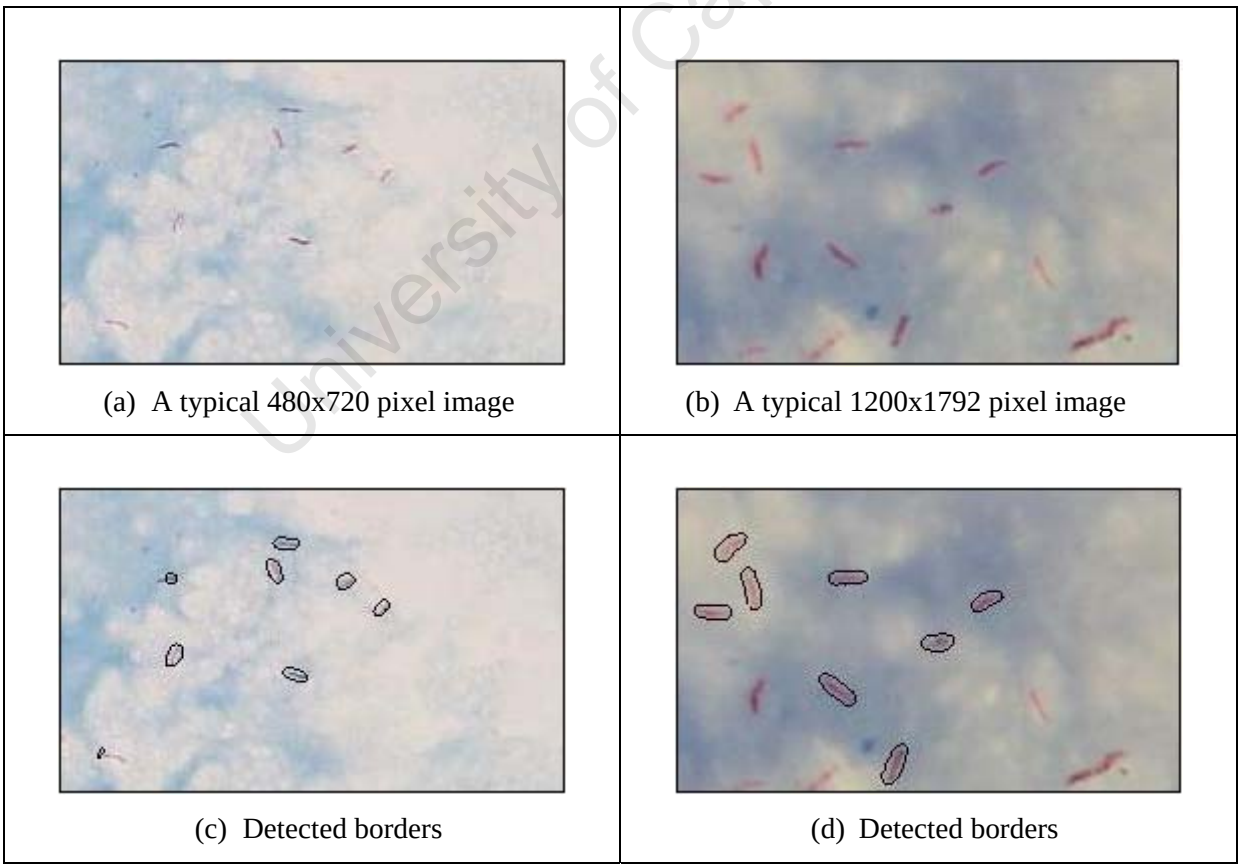


Figure 7.15: A image of a typical thick smear; arrow indicates a bacillus lying deep in the smear.

7.3 Parametric Deformable Model-based Segmentation

Figures 7.16 and 7.17 show typical bacillus borders detected by parametric deformable models.



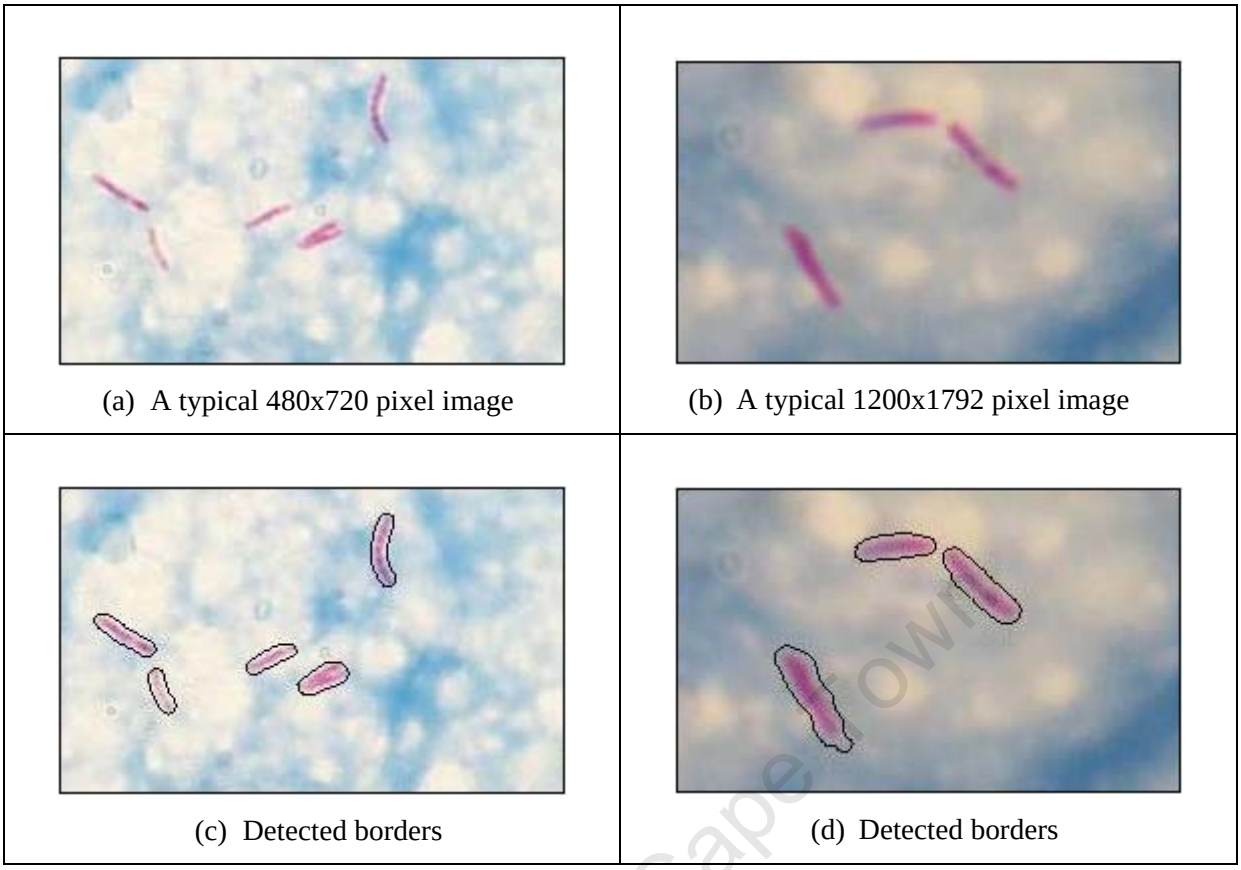
(a) A typical 480x720 pixel image

(b) A typical 1200x1792 pixel image

(c) Detected borders

(d) Detected borders

Figure 7.16: Typical borders detected by the parametric model in images captured at 40-x magnification.



(a) A typical 480x720 pixel image

(b) A typical 1200x1792 pixel image

(c) Detected borders

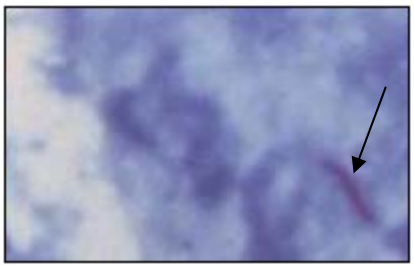
(d) Detected borders

Figure 7.17: Typical borders detected by the parametric model in images captured at 100-x magnification.

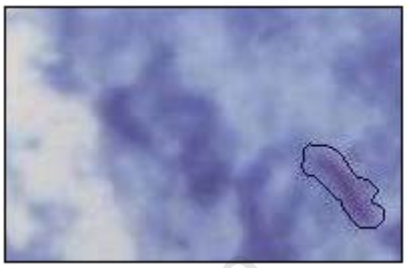
The parametric deformable model captures some non-red pixels around the bacilli, indicating that the algorithm under-segments the bacilli; this may be attributed to the blurring effect of the Gaussian filter that is applied during the computation of the external force. This effect may be reduced by using a smaller size for the Gaussian kernel; the value used by the algorithm in this implementation is 20 pixels. The value for the kernel size was determined experimentally by trial and error; smaller values tended to cause the model to over-segment some bacilli and for bacilli with a faint stain, edges may become so weak that the model leaks through the border and may fail to detect the object boundary.

The inclusion of non-red pixels may negatively affect the performance of a colour-based classifier; the contours may not adequately represent the shape of bacilli for shape-based classification. A boundary tracing stage for segmented objects may be necessary for such a classifier in order to reduce or eliminate the non-red pixels.

The performance of the algorithm depends on the quality of the stain; the algorithm performed better on images of well-stained smears with low bacillus density; performance was reduced in images of badly-stained samples as illustrated by Figure 7.18, which shows that the model stops far away from the actual border of the bacillus.



(a) Typical image of badly-stained smear



(b) Detected border

Figure 7.18: Performance of the parametric model on a typical image of a badly-stained smear; arrow indicates a bacillus object.

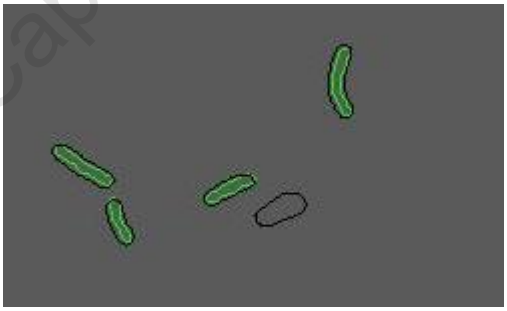
A major limitation of the algorithm is the long processing times. The processing time increases with image size due to an increase in the amount of data that has to be processed during evolution. The parametric model is also susceptible to self-intersections, particularly in images with high bacillus density; objects that cause intersections were discarded.

Table 7.3 summarises the differences between manually drawn borders and borders determined by the parametric deformable model algorithm for objects in 10 images of each set. On a computer screen, differences of 1-3 pixels are visually undetectable. Figure 7.19 illustrates the typical differences between manually drawn borders and parametric model-determined borders in a 100-x image with 480x720 pixel resolution.

Table 7.3: Mean Hausdorff distances and processing times for the parametric deformable model-based segmentations

Image set		Hausdorff distance (pixels)		Processing time per bacillus (seconds)	
Magnification	**Resolution (pixel)**	**Mean**	**Standard deviation**	**Mean**	**Standard deviation**
40-x	480 x 720	4.2	1.1	11.7	4.5
	1200 x 1792	8.5	5.0	83.0	16.2
100-x	480 x 720	6.7	1.8	16.4	6.9
	1200 x 1792	8.6	2.7	93.8	36.6

(a) Example image



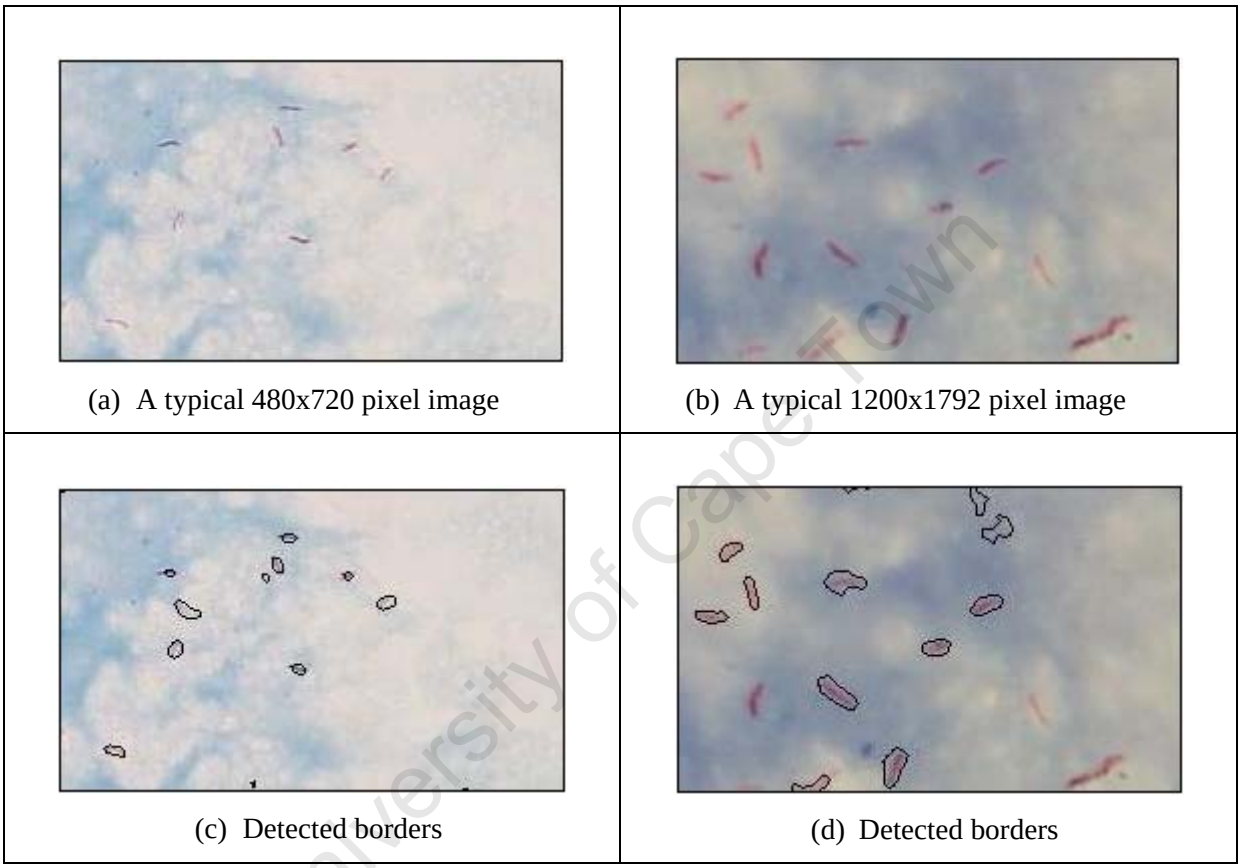
(b) Borders

Figure 7.19: Typical differences between parametric model-determined borders (shown in black) and manually drawn borders (shown in green) in 100-x, low-resolution images; arrow indicates clumped bacilli.

7.4 Geometric Deformable Model-based Segmentation

Results of the geometric model algorithm show that its performance is similar to that of the parametric model. Like with the parametric model, some non-red pixels are captured with the geometric model; this may also be attributed to the effect of convolving an image with a Gaussian filter during the computation of the edge indicator function. The algorithm performed better on images of well-stained samples.

The processing times for the geometric model were much longer than those for the parametric model. In addition to longer processing times, the geometric model also required more memory for operation and despite cropping the image under analysis around a target object to reduce the amount of data that is processed during contour deformation, the memory requirements still remained high. Figures 7.20 and 7.21 show typical borders obtained using the geometric model-based segmentation.



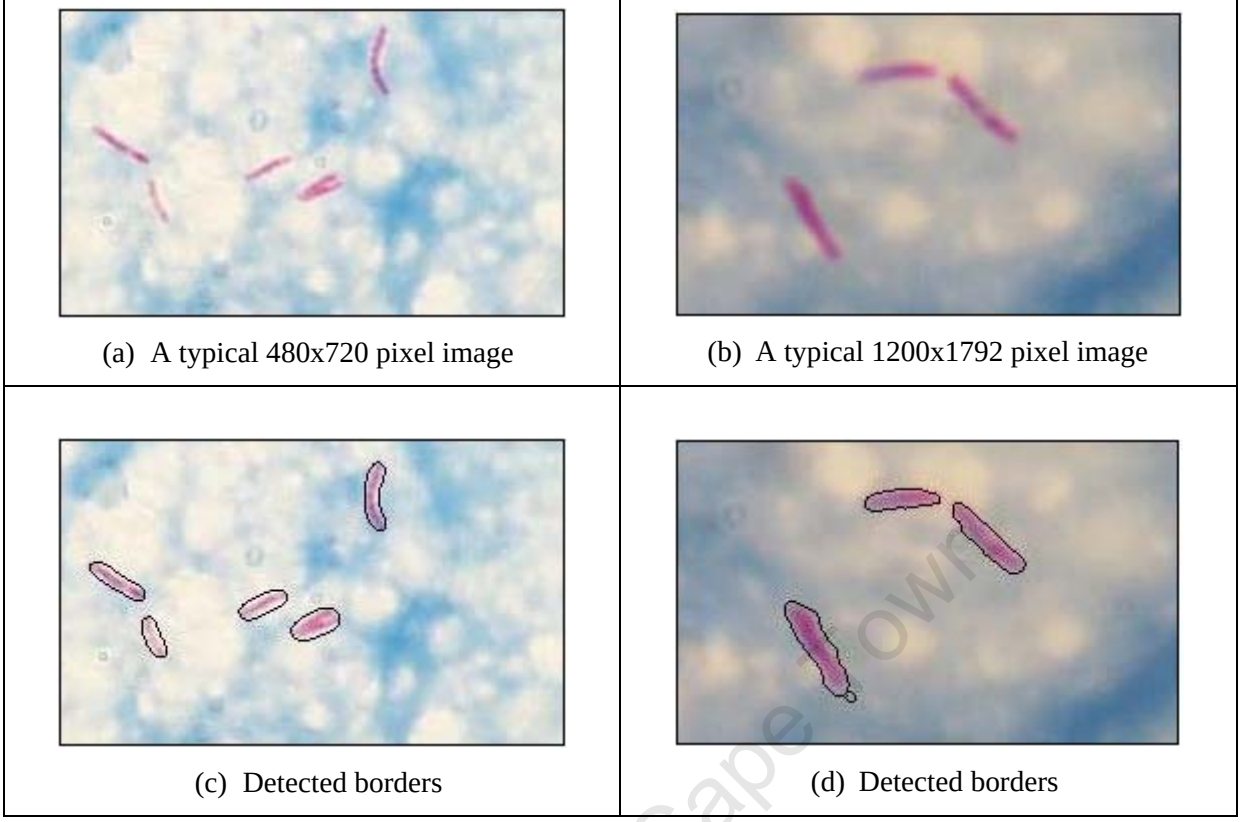
(a) A typical 480x720 pixel image

(b) A typical 1200x1792 pixel image

(c) Detected borders

(d) Detected borders

Figure 7.20: Typical borders detected by the geometric model in images captured at 40-x magnification.



(a) A typical 480x720 pixel image

(b) A typical 1200x1792 pixel image

(c) Detected borders

(d) Detected borders

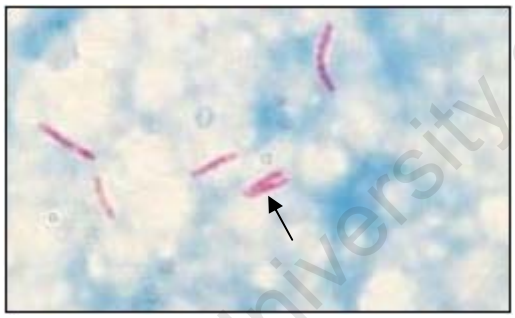
Figure 7.21: Typical borders detected by the geometric model in images captured at 100-x magnification.

Table 7.4 summarises the validation results of the algorithm. The average Hausdorff distances obtained for the geometric model are slightly smaller than those obtained for the parametric model; this is because the Gaussian kernel used for the geometric model has a size of 10 pixels compared to 20 pixels for the parametric model. Computation of the edge indicator function for the geometric model therefore produces less blurring resulting in a smaller shift of the actual boundary.

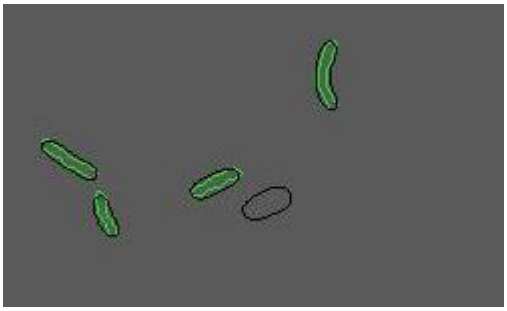
Table 7.4: Mean Hausdorff distances and processing times for the geometric deformable model-based segmentations

Image set		Hausdorff distance (pixels)		Processing time per bacillus (seconds)	
Magnification	**Resolution (pixel)**	**Mean**	**Standard deviation**	**Mean**	**Standard deviation**
40-x	480 x 720	3.7	1.4	14.5	8.7
	1200 x 1792	7.9	4.7	122.5	36.0
100-x	480 x 720	6.5	1.7	15.0	5.2
	1200 x 1792	8.1	3.1	291.2	75.0

Figure 7.22 illustrates the typical differences in borders determined by the geometric model algorithm and the manually-determined borders for an image captured at 100-x magnification with 480x720 pixel resolution.



(a) Example image



(b) Borders

Figure 7.22: Typical differences between geometric model-determined borders (shown in black) and manually drawn borders (shown in green) in 100-x, low-resolution images; arrow indicates clumped bacilli.

7.5 Validation of Initial Object Detection Method

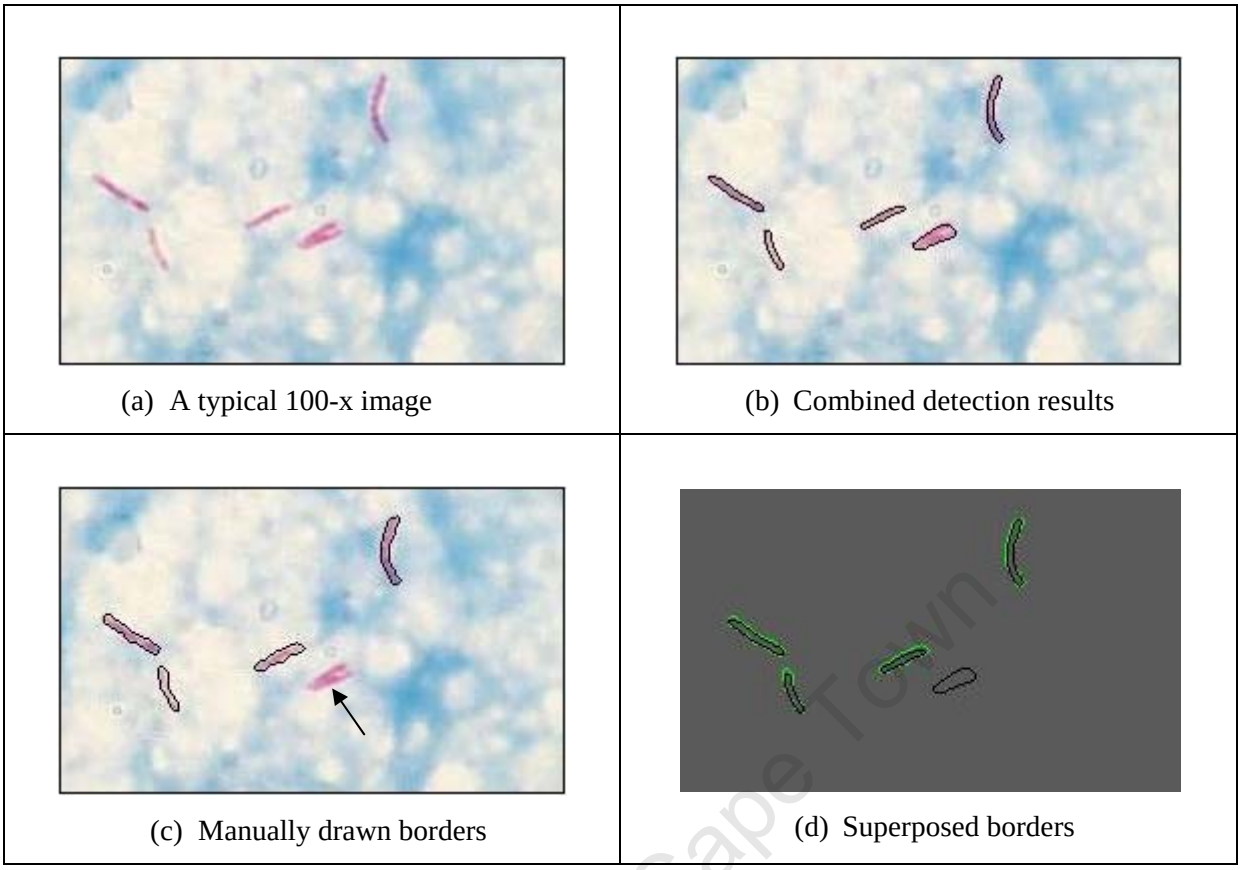
Validation of the combined results of the initialisation methods was carried out to assess the accuracy with which bacilli can be segmented by the method; this was done for the 480x720

pixel resolution images only since detected contours at this resolution do not have any gaps. Table 7.5 summarises validation results in 10 images of each set.

Table 7.5 Mean Hausdorff distances for the combined object detection method

Hausdorff Distance (pixels)	Image Set	
	40-x	**100-x**
Mean	3.1	3.4
Standard deviation	0.8	1.3

The validation results of the combined initialisation method show that the accuracy of this method is higher than that of the deformable models, particularly for the 100-x images. This method differs from the deformable model-based segmentation in that detected borders are inside the manually detected borders while those detected by deformable model-based segmentation are outside. In other words, relative to manual segmentation, the combined initialisation method over-segments the bacilli, i.e. the method excludes some red pixels that are part of bacilli as illustrated in Figure 7.23 where the bacilli segmented by the combined initialisation method have smaller widths than those segmented manually.



(a) A typical 100-x image

(b) Combined detection results

(c) Manually drawn borders

(d) Superposed borders

Figure 7.23: Typical differences between combined detection-determined borders (shown in black) and manually drawn borders (shown in green) in 100-x, low-resolution images; arrow indicates clumped bacilli.

7.6 Discussion

Objects that have been segmented may be used in a classification algorithm to determine whether or not they are indeed bacilli. Features of segmented objects can be extracted and used for classification. TB bacilli are known to exhibit a rod-like shape (Collins et al., 1985) and they show up with a reddish colour in images of ZN-stained smears; shape and colour are useful features that can be used to classify segmented objects. Shape descriptors such as Fourier descriptors and compactness may be computed for each segmented region and used in a shape-based classifier. The typical dimensions of bacilli can also be used as classification features. Clumped bacilli may lose the features that are characteristic of typical bacilli and a classifier may not recognise them as bacilli.

The parametric and geometric deformable models capture the shape of bacilli in images captured at 40-x magnification less well than they capture the bacillus shapes in 100-x

images. The performance of a shape-based classifier may be reduced if such a classifier is based on results from 40-x images. The occurrence of self-intersections of the parametric model is also increased in 40-x images with 480x720 pixel resolution.

Considering the fact that differences of 1-3 pixels are not visually detectable, the accuracy of both models may be considered similar. The parametric model, however, may be more useful in automated TB detection because of shorter processing times and lower memory requirements, depending on the success of a classifier in classifying objects detected by each method.

Deformable models generally produce under-segmented regions; for this reason, a boundary tracing stage, in which the inner boundary of a segmented region is traced and information about the position of the boundary pixels is extracted, may be necessary before classification of objects segmented by deformable models. This would help to reduce the number of non-red pixels in the detected regions and reduce the chances of misclassification in a colour-based classifier.

The combined object detection method has proven that it can detect bacilli in low-resolution images to produce borders with no gaps. While the method over-segments the bacilli, it does, however, capture the shape of the bacilli in images captured at 40-x and 100-x magnification. A colour-based classifier would be expected to correctly classify objects detected by this method since it does not capture any non-red pixels; a shape-based classifier would also be expected to produce good results with such objects since the object shapes appear to be adequately captured. Taking these factors into account, and that the run times for deformable model-based segmentation are long, the combined detection method would be the more efficient segmentation method for low-resolution sputum images. The combined object detection method has the added advantage that it does not have the high memory requirements and self-intersections that are associated with deformable models. With bacillus detection rates of 98.2 % and 99.9 % in 40-x and 100-x images respectively, the method maintains high detection rates; ultimately, classifier performance would determine its usefulness.

8. CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The aim of the project was to explore the use of deformable models for segmentation of images of sputum smears stained by the Ziehl-Neelsen method. The objectives of the study were to develop algorithms to automatically segment TB bacilli from sputum smear images using parametric and geometric deformable models and to assess the accuracy with which the TB bacilli could be segmented. Algorithms for a parametric and a geometric deformable model were successfully developed and the accuracy of segmentation by the two methods was evaluated by comparison with manual segmentation, using the Hausdorff distance.

The deformable model algorithms can segment objects in images to produce contours that are complete regardless of image size. The best results were obtained in images captured from well-stained samples in which the TB bacilli are distinct from the background. Other methods for formulating parametric and geometric deformable models should be explored before a final conclusion regarding image quality can be made.

The use of deformable models for segmentation of sputum smear images is limited by lengthy processing times and under-segmentation of bacilli. Parametric deformable models are also limited by occasional self-intersections. Geometric deformable models are further limited by computational complexity and high memory requirements.

Validation results showed that both types of models can detect object borders with similar accuracy; a more concrete conclusion may be reached by measuring the success of a classification algorithm for objects detected by each method. Both methods captured the shape of bacilli better in 100-x images than in 40-x images; bacillus shape was particularly poor in 40-x images with low resolution. While the processing times for the algorithms were lower for 40-x images, contours detected by the deformable models for objects in those images were circular; such contours were more common in the 480x720 pixel-resolution images. A shape-based classification algorithm may not work well for objects in images captured at 40-x magnification.

To achieve automatic initialisation, colour thresholding, Canny edge detection and watershed segmentation were used to locate objects in images. Combining the results of the three methods increased the number of detected objects. Detected objects in high-resolution images

had broken borders but borders for detected objects in low-resolution images were complete. The results of the method used for initial segmentation were validated for low-resolution images and the validation indicated that the technique of combining colour thresholding, Canny edge detection and watershed segmentation produces objects that may be input to a classifier without the application of deformable models. The initial segmentation method, unlike the deformable models, can sufficiently capture the shapes of bacilli in the 40-x images and would be more useful in an automated microscopy system that uses a 40-x objective lens.

Elimination of non-bacillus objects prior to deformable model-based segmentation was motivated by the need to reduce the processing time per image and also to prevent self-intersections of the parametric model caused by large objects. Objects were eliminated based on area and colour. The elimination method was successful at retaining possible bacillus objects.

8.2 Recommendations

A major limitation of using deformable models for segmenting sputum images is the long processing times; geometric models are further limited by high memory requirements. These limitations may be addressed by implementing the algorithms in a more efficient programming language that provides better memory utilisation such as C++ and/or in a parallel processing environment. The technique of cropping an image around a target object before implementation of the geometric model, can be applied to the parametric model algorithm too; this may considerably decrease the processing time for the parametric model.

Other formulation techniques for deformable models suggested in the literature should be explored; deformable models formulated by methods other than those used in this project might yield faster processing times. The technique proposed by Li et al. (2005) that allows automatic initialisation for parametric deformable models may be useful.

The initial detection of objects needs to be improved in order to increase the number of candidate bacillus objects. Further processing of results of the Canny edge detector to convert detected edges to closed contours defining objects would be useful.

The performance of the algorithms described in this thesis highly depend on the staining quality of a smear in the image. Preparation of samples should be improved and consistency in the quality of the stains should be maintained for the algorithms to be successful.

REFERENCES

Bowyer KW. Validation of Medical Image Analysis Techniques. In: Sonka M and Fitzpatrick JM, editors. *Handbook of Medical Imaging. Volume 2*. USA: Press Monograph, 2000; 567-607.

Brigger P, Hoeg J, Unser M. B-spline snakes: A Flexible Tool for Parametric Contour Detection, *IEEE Transactions on Image Processing*. 2000; 9(9): 1484-1496.

Caselles V, Catte F, Coll T, Dibos F. A Geometric Model for Active Contours, *Numerische Mathematik*. 1993; 66: 1-31.

Caselles V, Kimmel R, Sapiro G. Geodesic Active Contours. *Proceedings of 5th International Conference of Computer Vision*, 1995; 694-695.

Castleman KR. *Digital Image Processing*. New Jersey: Prentice-Hall; 1996.

Castleman KR. Concepts in Imaging and Microscopy: Colour Image Processing for Microscopy, *The Biological Bulletin*. 1998; 194(2):100-107.

Chakraborthy A, Staib LH, Duncan JS. Deformable Boundary Finding in Medical Images By Integrating Gradient and Region Information, *IEEE Transactions on Medical Imaging*, 1996; 15: 859-870.

Chan T, Vese L. Active Contours Without Edges, *IEEE Transactions on Image Processing*, 2001; 10: 266-277.

Chunming Li. c2005. Available from: http://www.engr.uconn.edu/~cmli/#Code, last accessed 3 February 2009.

Cohen LD. On Active Contour Models and Balloons, Computer Vision, Graphics and Image Processing: *Image Understanding*. 1991; 53 (2): 211-218

Cohen LD, Cohen I. Finite Element Methods for Active Contour Models and Balloons for 2-D and 3-D images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1993; 15(11): 1131-1147.

Collins CH, Grange JM, Yates MD. *Organisation and Practice in Tuberculosis Bacteriology*. London: Butterworths; 1985.

De Boor C. *A Practical Guide to Splines*. New York: Springer-Verlag; 1978.

Fenster A, Ladak H, Ding M. Deformable Model-based Segmentation of The Prostate from Ultrasound Images. In: Suri JS and Farag AA, editors. *Deformable Models II: Theory and Biomedical Applications*. New York: Springer Science + Business, 2007; 325-369.

Forero M, Sroubek F, Cristobal G. Identification of Tuberculosis Bacteria Based on Shape and Color, *Real Time Imaging*. 2004; 10(4): 251-262.

Gao J, Kosaka A, Kak A. A Deformable Model for Human Organ Extraction, *Proceedings of International Conference on Image Processing*, 1998; 323-327.

Gonzalez RC, Woods RE. *Digital Image Processing*. New Jersey: Prentice-Hall; 2002.

Gonzalez RC, Woods RE, Eddins SL. *Digital Image Processing Using MATLAB*. New Jersey: Prentice-Hall; 2004.

Han X, Xu C, Prince JL. A Topology Preserving Level Set Method for Geometric Deformable Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2003; 25(6): 755-768.

Hanscheid T. The Future Looks Bright: Low-cost Fluorescent Microscopes for Detection of Mycobacterium Tuberculosis and Coccidiae, *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 2008; 102: 520-521.

Harris A, Maher D, Graham S. *TB/HIV: A Clinical Manual*. World Health Organisation, Geneva, Switzerland. 2004.

Huttenlocher D. P, Klanderman G. A, Rucklidge W. J. Comparing Images Using the Hausdorff Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993; 15(9): 850-862.

Ihlow U, Seiffert U. Microscope Color Image Segmentation for Resistance Analysis of Barley Cells Against Powdery Mildew, *9 Workshop "Farbbildverarbeitung", ZBS Zentrum für Bild- und Signalverarbeitung eV Ilmenau, Report Nr 3/2003* 2003; 59–66.

Image Analysis And Communications Lab. c1999. Available from: http://www.iacl.ece.jhu.edu/resources/, last accessed 3 February 2009

Image Processing Place. c2004. Available from: http://www.imageprocessingplace.com/DIPUM_Toolbox_1/dipum_toolbox_main_page.htm, last accessed 3 February 2009

Jacob M, Blu T, Unser M. Efficient Energies and Algorithms for Parametric Snakes, *IEEE Transactions on Image Processing*. 2004; 13(9): 1231-1244.

Kass M, Witkin A, Terzopoulos D. Snakes: Active Contour Models, *International Journal of Computer Vision.* 1987; 1(4): 321–331.

Kovacevic D, Loncaric S, Sorantin E. Deformable Contour Based Method For Medical Image Segmentation, *Proceedings of 21st International Conference on Information Technology Interfaces*. 1999: 99-106.

Li C, Xu C, Gui C, Fox MD. Level Set Evolution Without Re-Initialization: A New Variational Formulation, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005

Li C, Liu J, Fox MD. Segmentation of Edge Preserving Gradient Vector Flow: An Approach Toward Automatically Initializing and Splitting of Snakes, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005; 1: 162 – 167.

Malladi R, Sethian J. A, Vemuri B. C. Shape Modelling With Front Propagation: A Level Set Approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1995; 17 (2): 158-175.

MATLAB Version 7.3.0, Help Notes on *bwboundaries.m*; 2006.

McInerney T, Terzopoulos D. Topology Adaptive Snake, *In 5th International Conference on Computer Vision*, Cambridge Massachusetts, 1995: 840-845.

McInerney T, Terzopoulos D. Deformable Models in Medical Image Analysis: A Survey, *Medical Image Analysis*, 1996; 1: 91-108

Menet S, Saint-Marc P, Medioni G. B-snakes: Implementation and Application to Stereo, *Proceedings of Image Understanding Workshop*. 1990; 720-726.

Najarian K, Splinter R. *Biomedical Signal and Image Processing*. Florida: Taylor & Francis; 2006

Nguyen TNL, Wells CD, Binkin NJ, Pham DL, Nguyen VC. The Importance of Quality Control of Sputum Smear Microscopy: The Effect of Reading Errors on Treatment Decisions and Outcomes, *International Journal of Tuberculosis and Lung Disease*. 1999; 3(6): 483-487.

Osher S, Sethian JA. Fronts Propagating With Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*. 1988; 79(1): 12-49.

Precioso F, Barlaud M, Blu T, Unser M. Smoothing B-Spline Active Contour for Fast and Robust Image and Video Segmentation. *IEEE Transactions on Image Processing*, 2005; 14(7): 910-923.

Ranganath S. Contour Extraction From Cardiac MRI Studies Using Snakes, *IEEE Transactions on Medical Imaging*, 1995; 14: 328-338

Ray N, Acton ST et al. Merging Parametric Active Contours Within Homogeneous Image Regions for MRI-based Lung Segmentation. *IEEE Transactions on Medical Imaging*, 2003; 22(2): 189-199.

Rogers DF, Adams JL. *Mathematical Elements for Computer Graphics*. New York: McGraw-Hill; 1990.

Russell M. *Autofocusing a Smart Microscope for the Detection of Tuberculosis in Sputum Smears* (MSc Dissertation). Cape Town (South Africa): University of Cape Town: 2004.

Sadaphal P, Rao J, Comstock GW, Beg MF. Image Processing Techniques for Identifying *Mycobacterium Tuberculosis* in Ziehl-Neelsen Stains, *International Journal of Tuberculosis and Lung Disease*. 2008; 12(5): 579-582.

Siddiqi K, Lauziere YB, Tannenbaum A, Zucker SW. Area and Length Minimising Flows for Shape Segmentation. *IEEE Transactions on Image Processing*, 1998; 7: 433-443.

Sonka M, Hlavac V, Boyle R. *Image Processing, Analysis and Machine Vision*. Toronto: Thompson; 2008.

South African National Tuberculosis Association, Latest Tuberculosis Statistics. 2004 http://www.santa.org.za/TBstatistics.html, last accessed 15 July 2008

Steingart KR, Henry M, Ng V, Hopewell PC, Ramsay A, Cunningham J et al. Flourescence Versus Conventional Sputum Smear Microscopy for Tuberculosis: A Systematic Review. *Lancet Infectious Diseases.* 2006; 6(9):570-81.

Todar K. Tuberculosis, *Todar's Online Textbook of Bacteriology.* 2008 http://www.textbookofbacteriology.net/tuberculosis.html, last accessed 14 July 2008

Velut J, Benoit-Cattin H, Odet C. Locally Regularised Smoothing B-Snake. *EURASIP Journal on Advances in Signal Processing*, 2007

Vemuri B, Chen Y. *Joint Image Registration and Segmentation. Geometric Level Sets In Imaging, Vision and Graphics*, 2003; 251-269.

Veropoulos K, Learmonth G, Campbell C, Knight B, Simpson J. Automated Identification of Tubercle Bacilli in Sputum: A Preliminary Study. *Analytical and Quantitative Cytology and Histology*, 1999; 21(4): 277-281.

Veropoulos K. *Machine Learning Approaches to Medical Decision Making* (PhD Dissertation). Bristol (UK): University of Bristol: 2001.

WHO fact sheet no.104, 2007. World Health Organisation, Geneva, Switzerland.

Wong KP. Medical Image Segmentation: Methods and Applications in Functional Imaging. In: Suri JS, Wilson DL, Laxminarayan S, editors. *Handbook of Biomedical Image Analysis: Volume 2: Segmentation Models, Part B*. New York: Kluwer Academic/Plenum Publishers, 2005; 111-182.

Xie X, Mirmehdi M. RAGS: Region-Aided Geometric Snake, *IEEE Transactions on Image Processing*. 2004; 13(5): 640-652.

Xu C, Prince J.L. Snakes, Shapes and Gradient Vector Flow, *IEEE Transactions on Image Processing*. 1998; 7(3): 359-369.

Xu C, Prince J.L. Generalized Gradient Vector Flow External Forces for Active Contours, *Signal Processing*. 1998; 71(2): 131-139.

Xu C, Pham D.L, Prince J.L. Medical Image Segmentation Using Deformable Models. In: Fitzpatrick JM and Sonka M, editors. *Handbook of Medical Imaging - Volume 2: Medical Image Processing and Analysis*. New York: Kluwer Academic/Plenum Publishers, 2000: 129-174

Xu C, Prince JL. Gradient Vector Flow Deformable Models. In: Bankman I, editor. *Handbook of Medical Imaging: Processing and Analysis*. Baltimore: Academic Press, 2000: 159-170.

Xu C, Yezzi A, Prince JL. On the Relationship between Parametric and Geometric Active Contours, *Proceedings of 34th Asilomar Conference on Signals, Systems, and Computers*, 2000: 483-489.

Yu Z, Bajaj C. Image Segmentation Using Gradient Vector Diffusion and Region Merging, *16th International Conference on Pattern Recognition*. 2002: Volume 2

BIBLIOGRAPHY

Giraldi GA, Rodrigues PS, Marturelli LS, Silva RLS. Improving the Initialisation, Convergence and Memory Utilisation for Deformable Models. In: Suri JS, Wilson DL, Laxminarayan S, editors. Handbook *of Biomedical Image Analysis: Volume 1: Segmentation Models, Part A. New York*: Kluwer Academic/Plenum Publishers, 2005: 111-182.

Jahne B. *Digital Image Processing: Concepts, algorithms and Scientific Applications*. Germany: Springer-Verlag Berlin Heidelberg; 1993.

Jain AK. *Fundamentals of Digital Image Processing*. USA: Prentice-Hall; 1989.

Marion A. *An Introduction to Digital Image Processing*. Great Britain: Prentice-Hall; 1991.

Niblack W. *An Introduction to Digital Image Processing*. Denmark: Prentice-Hall; 1986.

Rusnell BJ, Pierson RA, Singh J, Adams GP, Eramian MG. Level Set Segmentation of Bovine Corpora Lutea in Ex-situ Ovarian Ultrasound Images, *Reproductive Biology and Endocrinology*. 2008; 33(6).

Selvakumar N, Murthy BN, Prabhakaran E et al., Lot Quality Sampling of Acid-fast Bacillus Smears for Assessing Sputum Smear Microscopy Centers, *Journal of Clinical Microbiology*. 2005; 43(20): 913-915.

Suri JS, Farag AA (Editors). *Deformable Models II: Theory and Biomaterial Applications*. New York: Springer Science + Business Media; 2007.

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*. 2004; 13(4): 600-612.

APPENDIX

Contents of CD

The CD supplied with this dissertation has three folders; Parametric, Geometric and Miscellaneous, whose contents are given below.

Parametric Folder

This folder contains the m-files for the parametric model-based segmentation and the following functions that are used by the algorithm:

- parametric.m – this is the main function for parametric model-based segmentation. To run the algorithm, type parametric at the prompt, with the magnification (40 or 100) and the resolution (480 or 1200 for low-resolution and high-resolution images respectively) in brackets, and press ENTER, select an image to be segmented using the dialogue box. The output of the algorithm is a cell array that contains the coordinates of the final position of the deformable model for each object. The algorithm displays the input image with the detected objects outlined.
- snake.m – this is a function that is called by the main function to evolve the deformable model.
- combined_detection.m – a function called by the main function to perform the initial segmentation for automatic initialisation of deformable models.
- covmatrix.m – a function used by the combined detection algorithm to compute the average colour of detected regions during colour-based elimination of objects.
- imstack2vectors.m – a function used by the colour thresholding algorithm.

Geometric Folder

This folder contains the m-file for the geometric model based segmentation. In order to improve memory utilisation, the m-file uses nested functions and hence there are no other functions in the folder. The m-file requires storage space to save some variables during operation and hence one needs to copy the m-file to a directory that has at least 50MB free space.

To run the algorithm, type geometric at the prompt, with the magnification (40 or 100) and the resolution (480 or 1200 for low-resolution and high-resolution images respectively) in brackets, and press ENTER, select an image for segmentation using the dialogue box. The

output of the algorithm is a cell array that contains the coordinates of the final position of the deformable model for each object. The algorithm displays the input image with the detected objects outlined.

Miscellaneous Folder

This folder contains the m-files that were used during the validation of the segmentations.

- manual.m – a function used for manually outlining bacilli. To run the algorithm , type manual at the command line and press ENTER, use the GUI that pops up to select an image for segmentation. Follow instructions given at the command line and save the results when done.
- manual.fig – a Graphic user interface figure, used by the manual.m file, that enables a user to select an image for manual segmentation.
- hausdorff.m – a function that computes the Hausdorff distance between two contours. To run the algorithm, load the files that contain the results to be validated at the command line and enter hausdorff at the command line; the output is a vector of Hausdorff distances. To validate results that contain multiple contours, make sure that the order of the manual contours correspond with the order of the automatic contours i.e. the k^{th} manual contour corresponds to the k^{th} automatic contour.
- computedist.m – a function, called by the hausdorff.m function, that computes the directed Hausdorff distances.
- object_detection.m – a function that performs and combines the results of colour thresholding, Canny edge detection and watershed segmentation, and displays the results. To run the algorithm, type object_detection at the prompt, with the magnification (40 or 100) and the resolution (480 or 1200 for low-resolution and high-resolution images respectively) in brackets, and press ENTER, select an image to be segmented using the dialogue box. The output of the algorithm is a cell array that contains the coordinates of the positions of the border pixels of detected objects. The algorithm displays the input image with the detected objects outlined.